

# **DISEÑO DE UN NUEVO CLASIFICADOR HÍBRIDO NO SUPERVISADO PARA TEXTURAS NATURALES EN IMÁGENES**

**MÁSTER EN INVESTIGACIÓN  
EN INFORMÁTICA  
2006-07**

**TRABAJO DE INVESTIGACIÓN**



Departamento de Ingeniería del Software e Inteligencia Artificial  
Facultad de Informática  
Universidad Complutense  
MADRID

Alumna:  
María Guijarro Mata-García

Directores:  
Gonzalo Pajares Martinsanz  
Luis Garmendia Salvador



# Resumen

En este trabajo se presenta el diseño de un nuevo clasificador híbrido para la clasificación de imágenes de texturas naturales. Existen diversos métodos de clasificación de texturas en la literatura especializada, siendo generalmente necesario el diseño de nuevos métodos con el fin de mejorar los resultados de la clasificación. Este es el objetivo del presente trabajo donde se presenta un nuevo clasificador que aprovecha las bondades de tres de los clasificadores clásicos existentes, a saber: Fuzzy, Bayes y Parzen. Mediante la combinación de estos tres métodos bajo los denominados operadores de agregación procedentes de la lógica fuzzy o borrosa se diseña el nuevo clasificador híbrido. Los resultados se comparan con los métodos clásicos anteriormente mencionados.

**Palabras clave:** clasificación, texturas, clasificador híbrido, lógica fuzzy, clasificador fuzzy, clasificador Bayesiano, clasificador de Parzen.

# Agradecimientos

A la empresa Digital Image Processing (Dimap) (<http://www.dimap.es/>) que nos ha proporcionado las imágenes para llevar a cabo la investigación, a su vez procedentes del Servicio Territorial de Galicia (SITGA).



# Índice

<b>1. INTRODUCCIÓN.....</b>	<b>1</b>
1.1. IDENTIFICACIÓN DEL PROBLEMA .....	1
1.2. OBJETIVOS DE INVESTIGACIÓN .....	4
1.2.1. Objetivos generales .....	4
1.2.2. Objetivos específicos.....	5
1.3. METODOLOGÍA HISTÓRICA .....	5
1.4. APORTACIONES A LA INVESTIGACIÓN .....	8
<b>2. REVISIÓN DE MÉTODOS .....</b>	<b>9</b>
2.1. MÉTODOS DE CLASIFICACIÓN DE TEXTURAS NATURALES .....	9
2.2. DESCRIPCIÓN DE LOS CLASIFICADORES SELECCIONADOS.....	12
2.2.1. Agrupamiento borroso.....	12
2.2.2. Clasificador Bayesiano.....	15
2.2.3. Ventana de Parzen .....	18
2.3. MÉTODOS CLÁSICOS DE COMBINACIÓN DE CLASIFICADORES .....	18
2.4. COMBINACIÓN MEDIANTE OPERADORES DE AGREGACIÓN .....	20
2.4.1. Aspectos generales de la teoría de agregación fuzzy .....	20
2.4.2. Tipos de operadores de agregación .....	21
<b>3. DISEÑO DE CLASIFICADOR HÍBRIDO NO SUPERVISADO .</b>	<b>27</b>
3.1. INTRODUCCIÓN.....	27
3.2. ANTECEDENTES .....	29
3.3. DISEÑO DEL MÉTODO HÍBRIDO.....	31
3.3.1. Fase de entrenamiento .....	31
3.3.2. Fase de clasificación.....	34
3.4. HIBRIDACIÓN.....	38
<b>4. ANÁLISIS DE RESULTADOS .....</b>	<b>45</b>
4.1. OBJETIVOS DEL ANÁLISIS .....	45
4.2. DESCRIPCIÓN DE LAS IMÁGENES UTILIZADAS .....	45
4.3. DISEÑO DE UNA ESTRATEGIA DE PRUEBA.....	48
4.4. ANÁLISIS DE RESULTADOS.....	51
4.4.1. Determinación del número de clases.....	51
4.4.2. Criterio de validación .....	52
4.4.3. Resultados obtenidos.....	54
<b>5. CONCLUSIONES .....</b>	<b>59</b>
5.1. SOBRE RESULTADOS .....	59
5.2. CUMPLIMIENTO DE LOS OBJETIVOS PROPUESTOS .....	60
5.3. TRABAJOS FUTUROS .....	61
5.4. DIVULGACIÓN DE RESULTADOS .....	61
<b>6. BIBLIOGRAFÍA .....</b>	<b>63</b>
<b>7. ANEXO I.....</b>	<b>67</b>
<b>8. ANEXO II .....</b>	<b>77</b>



# Capítulo 1

## 1. Introducción

### 1.1. Identificación del problema

La cada vez más sofisticada tecnología de los dispositivos de captura de imágenes hace que el interés por la utilización de las imágenes en diferentes aplicaciones sea cada día mayor. En efecto, hasta no hace muchos años las aplicaciones basadas en imágenes eran relativamente escasas, particularmente en el mundo empresarial. Actualmente esta situación está cambiando principalmente debido a una serie de factores, que citamos a continuación:

- 1) Los dispositivos de captura industriales se fabrican con altas resoluciones
- 2) Los satélites artificiales proporcionan imágenes cuyas resoluciones están llegando actualmente hasta observaciones de 20 ó 30 cm.
- 3) Los vuelos aéreos permiten obtener imágenes con resoluciones incluso mayores que las mencionadas anteriormente.
- 4) Los vehículos aéreos no tripulados (“UAVs: Unmanned Aerial Vehicles”) se están equipando con sensores de captura de imágenes de alta resolución.

Todo lo anterior hace que exista un interés creciente por la utilización de las más modernas tecnologías en el campo del tratamiento de las imágenes para diversas aplicaciones. Para dar idea de su utilidad mencionamos tan sólo algunas de tales aplicaciones:

- 1) Control de cultivos en agricultura, propiciado por la necesidad de conocer los cultivos programados para la recepción de subvenciones u otros aspectos relacionados.
- 2) Cómputo y medición de parcelas agrícolas y tipo de cultivos a los que se dedica.
- 3) Control de riegos agrícolas
- 4) Agricultura reprecisión para aplicar herbicida en el tratamiento de malas hierbas de forma selectiva evitando la contaminación medioambiental y la reducción de costes de producción.

- 5) Evaluación de catástrofes naturales: fuegos, daños por inundaciones, heladas en cultivos agrícolas, nevadas, etc.
- 6) Detección de cambios en determinadas zonas, principalmente urbanas para el control de edificaciones o impactos medioambientales
- 7) Vigilancia en prevención de catástrofes, por ejemplo fuegos o inundaciones
- 8) Control de fenómenos meteorológicos como es el retroceso de determinadas playas
- 9) Vigilancia: forestal, marítima.
- 10) Detección de infraestructuras: carreteras, caminos forestales, cañadas reales, etc.

En la figura 1.1 se muestra un ejemplo de clasificación de texturas cuya finalidad consiste en la identificación de las texturas naturales para realizar una planificación de caminos de cara a una intervención ante posibles catástrofes. Se trata de un ejemplo de simulación actualmente en fase de desarrollo dentro del grupo de investigación ISCAR (reconocido por la UCM) al que pertenece uno de los directores de la investigación. En (a) se muestra un mapa del entorno visto por un UAV. En (b) la clasificación de texturas realizada por un clasificador automático. En (c) un camino desde la base a un hipotético objetivo ante la posibilidad de que se trate de una intervención de urgencia por algún motivo del tipo de catástrofe de incendio o accidente.

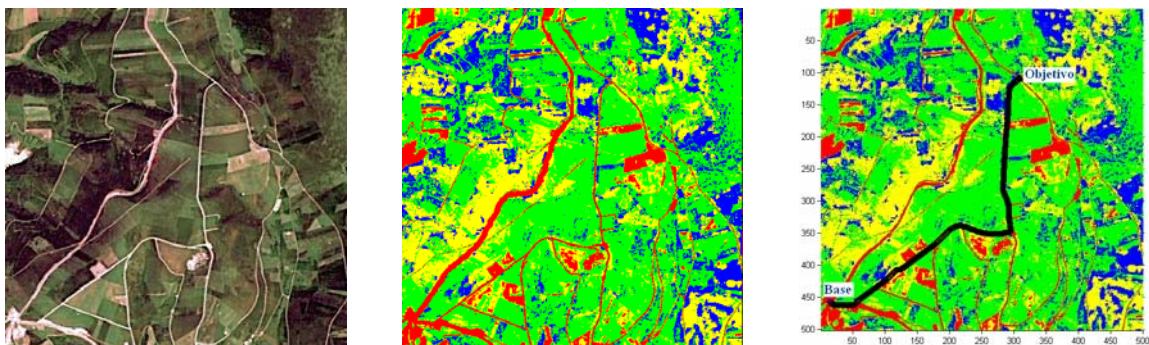


Figura 1.1 Ejemplo de aplicación de las texturas naturales

Se deduce de lo expuesto que en estas aplicaciones, el tratamiento de las **texturas naturales** en las imágenes juega un papel primordial.

En este sentido, diversas empresas u organismos desarrollan o utilizan aplicaciones para abordar dicha problemática. Por citar sólo algunas podemos mencionar

- 1) Digital Image Processing (Dimap) (<http://www.dimap.es/>)



- 2) Proespacio (<http://www.proespacio.org/>) agrupación de empresas del sector aeroespacial donde una de las actividades destacables son aplicaciones mediante el uso de imágenes de satélite. En este consorcio destacan algunas empresas líderes del sector tanto en España como a nivel Internacional: EADS-Espacio, EADS Astrium, CRISA, GMV, Indra Espacio, Sener, Hispasat, IberEspacio, Inasmet, Insa, Mier, NTE, Tecnológica, Ryma, Hispasat, GTD, Alcatel, CRISA, GTD.
- 3) Organismos oficiales y Centros de Investigación
  - Consejo Superior de Investigaciones Científicas (CSIC)
  - Instituto Nacional de Técnica Aeroespacial (INTA)
  - Centro de Estudios y Experimentación de Obras Públicas (CEDEX), con el que existen trabajos de colaboración previos por parte de uno de los directores del trabajo (Pajares y col. 2001, Pajares y col. 2002)

La mayoría de las empresas e instituciones anteriormente mencionadas utilizan para llevar a cabo sus aplicaciones principalmente herramientas comerciales tales como:

- ERDAS
- Intergraph
- ENVI-IDL
- ILOG
- PCI
- E-Cognition

Aunque bien es cierto que cada día es mayor la potencialidad de las herramientas citadas, no es menos cierto que los retos tecnológicos derivados de las aplicaciones mencionadas, hacen que en algunos casos la utilización de tales herramientas sea insuficiente para abordar las propuestas de proyectos demandados por los clientes. Este es el caso para una gran parte de las aplicaciones mencionadas en el tratamiento de las texturas naturales, donde la **clasificación** de las mismas surge como una tarea fundamental.

En efecto, en general las mencionadas herramientas implementan algunos de los métodos clásicos de clasificación, siendo necesaria la intervención del usuario mediante programación para abordar ciertas tareas que involucran aspectos de clasificación de

texturas, lo cual no siempre es factible o al menos en la medida que cabría esperar debido a múltiples limitaciones.

Además, y lo que es más importante, en muchos casos no existe la posibilidad de llevar a cabo la investigación necesaria para abordar la problemática, particularmente cuando los métodos clásicos no producen los resultados esperados.

Por todo lo expuesto anteriormente surge una necesidad importante en el ámbito de las aplicaciones reales para abordar el tema de la clasificación de texturas naturales en imágenes y un reto para la comunidad científica para tratar de mejorar los procedimientos existentes con la mayor flexibilidad posible.

La presente memoria se organiza en cinco capítulos. En lo que resta del presente capítulo se exponen los objetivos que se plantearon en la investigación aquí recogida, en base a los planteamientos previos y que tratan de dar solución a la problemática existente. También se exponen las estrategias realizadas para abordarlos. En el capítulo 2 se realiza una revisión del estado del arte en el tema de clasificación de texturas. En el capítulo 3 se propone la estrategia de solución a la problemática planteada, junto con una breve revisión de los métodos empleados para llevarla a cabo. En el capítulo 4 se diseña una estrategia de pruebas con el objetivo de verificar la eficacia del método propuesto cuando se compara con otras estrategias existentes. Finalmente en el capítulo 5 se extraen las conclusiones pertinentes y se exponen las líneas de investigación futuras, que dan pie al inicio de la investigación en los estudios de Doctorado.

## **1.2. Objetivos de investigación**

### **1.2.1. Objetivos generales**

- 1) Aprender a manejar referencias bibliográficas, así como la forma de abordar las investigaciones
- 2) Identificar métodos de clasificación de texturas existentes en la literatura
- 3) Determinar los métodos más utilizados
- 4) Analizar las ventajas e inconvenientes de cada uno de ellos así como su problemática
- 5) Dar solución a un problema real de aplicación empresarial realizando un aporte de carácter investigador.

### **1.2.2. Objetivos específicos**

- 1) Seleccionar al menos dos métodos de clasificación de entre los más prometedores o apropiados que se hayan estudiado
- 2) Seleccionar las imágenes que constituirán los conjuntos de entrenamiento y de clasificación
- 3) Identificar una posible mejora de los métodos existentes o dar una solución novedosa a la problemática de la clasificación de las texturas naturales.
- 4) Implementar la mejora propuesta con el fin de analizar su comportamiento en base a los resultados obtenidos.
- 5) Identificar líneas de investigación futuras
- 6) Difundir los resultados mediante publicaciones científicas u otro tipo de divulgación

## **1.3. Metodología histórica**

A continuación se expone de forma cronológica los pasos seguidos para llevar a cabo la investigación junto con la actividad desarrollada en cada uno de ellos.

Se inicia el trabajo de investigación dentro de la empresa Digital Image Processing (Dimap) (<http://www.dimap.es/>), donde la clasificación de texturas aparece como un reto para dar respuesta a diversas peticiones de clientes entre los que destaca Indra Sistemas para aplicaciones de simulación de vuelos aéreos fundamentalmente. Es en el ámbito empresarial donde se identifica la problemática.

Como consecuencia de ello se inicia el estudio de los métodos que tienen implementados algunas herramientas comerciales.

Ya dentro del programa Master, y en base a la problemática identificada a la que hay que proporcionar una solución, se identifican algunos de los cursos, que en principio parecen más acordes con el tema objeto de investigación. Este hecho hace que se parta de una posición favorable para la investigación ya que se tiene un problema y un objetivo claro. Los cursos seleccionados y realizados durante el Master son los siguientes:

- Percepción Visual Artificial (primer cuatrimestre)
- Razonamiento Aproximado y con Incertidumbre (primer cuatrimestre)
- Agentes Inteligentes (primer cuatrimestre)
- Redes Neuronales Artificiales (segundo cuatrimestre)
- Control Inteligente (segundo cuatrimestre)
- Ingeniería Lingüística (primer cuatrimestre)

En percepción Visual se aborda el tratamiento de las imágenes digitales desde el punto de vista del reconocimiento y el aprendizaje. Comienzan a sentarse las bases de la investigación desde el inicio del curso, que culmina con un trabajo sobre reconocimiento de patrones en el ámbito de las texturas naturales.

La línea anterior tiene su continuidad en el segundo cuatrimestre con Redes Neuronales y Control Inteligente. En la primera se continúa el estudio de métodos de clasificación mientras que en la segunda se abordan técnicas muy próximas a los trabajos del curso sobre Razonamiento Aproximado. En este caso se desemboca en el estudio de la clasificación desde el punto de vista de la lógica borrosa. En el intento de automatizar el razonamiento y el aprendizaje resulta muy útil dicha lógica.

Las teorías de razonamiento aproximado e inferencia borrosa están siendo muy utilizadas ya que son diversos los contextos en los que se debe obtener información útil a partir de datos incompletos, imprecisos o inciertos. El ser humano puede razonar y tomar decisiones a partir de información que raramente es precisa y que muchas veces puede ser modelada por generalizaciones del modus ponens clásico.

Dentro del ámbito de la lógica borrosa se estudian métodos de agregación borrosa, consistentes en fusionar la información proporcionada por varios métodos.

En Agentes Inteligentes se desarrolla un sistema de agentes cooperativos basados en sendos sistemas, uno de ellos es un UAV como integrante de un equipo de intervención ante catástrofes, que tiene como tarea inteligente la realización de un mapa del entorno a partir de las imágenes de texturas captadas mediante un sistema de visión artificial. La idea proviene del trabajo de investigación en el que participa el profesor D. Gonzalo Pajares con la empresa Indra Sistemas, S.A. y cuyo proyecto se denomina: “Metodología de

Modelado y Simulación para los entornos distribuidos”, se trata de un proyecto de ámbito Europeo en el que participan las siguientes empresas de vanguardia en el campo de la Simulación en Europa constituyentes del consorcio: DUTCH SPACE (NL), DATAMAT (IT), HAGGLUNDS(S), IFAD (DK), INDRA (ES), INSTA (FI), KDA (N), Patria (FI), S.I.A. (IT), TNO (NL).

En resumen, se estudian los siguientes métodos de clasificación como potencialmente viables. Se eligen en función de su probada eficacia en múltiples aplicaciones en la literatura:

- Clasificador paramétrico Bayesiano
- Clasificador no paramétrico: ventana de Parzen
- Agrupamiento borroso o Fuzzy clústering
- Algoritmos competitivos: Cuantización vectorial (“Learning Vector Quantization”), Mapas Auto-organizativos (Self-Organizing Feature Maps)
- Red neuronal retropropagación (“backpropagation”)

De entre todos ellos, los tres primeros son los que mejores resultados producen, por lo que se opta por seleccionarlos para el trabajo de investigación como candidatos a mejorar su rendimiento desde el punto de vista de eficacia en el porcentaje de aciertos.

Se investiga sobre métodos que utilicen varios clasificadores, encontrándose la *combinación de clasificadores* como el método que aborda esta problemática. Destacan métodos ampliamente utilizados en la literatura tales como: “majority voting”, máximo o media (Kittler y col. 1998).

En el caso que nos ocupa no se aprecia una mejora sustancial de los resultados obtenidos con las combinaciones anteriormente mencionadas respecto de los obtenidos con cada uno de ellos de forma individual. Es en este momento cuando surge la necesidad de buscar nuevas soluciones. Una de tales soluciones nos la proporciona la lógica borrosa a través de la *agregación de operadores*, ya que es posible combinar varios operadores con el fin de poder fusionar la información mediante la agregación mencionada. Es aquí donde se produce un aporte importante a la investigación en el campo de la clasificación de las

texturas naturales. Los términos *borroso* y *fuzzy* se utilizarán indistintamente a lo largo de esta memoria.

Queda todavía por resolver un aspecto relacionado con el procedimiento de clasificación automático que se pretende desarrollar. Se trata de que los métodos de clasificación seleccionados tienen el inconveniente de que son de naturaleza supervisada (Bayes, Parzen y Fuzzy clustering). Esto obliga a proporcionar inicialmente una serie de clases junto con las muestras asociadas a cada clase. Es entonces cuando se pierde la naturaleza automática del método propuesto para quedarse en un procedimiento semi-automático en el sentido de que es preciso realizar una inicialización de forma dirigida, esto es, supervisada. Con tal propósito se investiga en esta línea proponiendo, a partir de la teoría existente un método totalmente automático de naturaleza no supervisada.

Los resultados así obtenidos se comparan con los que proporcionan los métodos individuales y también con los derivados de la combinación de clasificadores antes mencionada. Se verifica que el método propuesto en este trabajo consigue unos resultados satisfactorios en el campo de la clasificación de texturas naturales, ya que mejoran a los anteriores.

## **1.4. Aportaciones a la investigación**

Como resumen de todo lo anterior las aportaciones de investigación realizadas en el presente trabajo son las siguientes:

- 1) Diseño de un clasificador no supervisado basado en tres métodos de clasificación clásicos
- 2) Combinación de los métodos de clasificación mediante operadores de agregación borrosa.
- 3) Estudio comparativo del método propuesto frente a otros métodos de clasificación combinados y los propios métodos aislados.

# Capítulo 2

## 2. Revisión de métodos

### 2.1. Métodos de clasificación de texturas naturales

Se han propuesto diferentes métodos de clasificación de texturas naturales en la literatura, entre los que destacan algunos tales como: el clasificador Bayesiano, Vecino más próximo (K-Nearest), Redes Neuronales (Pajares y col. 2001) o Cuantización vectorial (Learning Vector Quantization) (Giacinto y col. 2000, Chan y col. 2003, del Frate 2007), Agrupamiento borroso (Pajares y Cruz, 2002, Pajares y col. 2002).

Un aspecto importante a destacar en el tema de la clasificación en general es el hecho de que la **combinación** de clasificadores obtiene mejores resultados que los clasificadores utilizados de forma individual (Valdovinos y col. 2005, Kuncheva, 2004, Kumar y col. 2002 o Kittler y col. 1998). En este sentido destacan dos importantes estrategias comúnmente aceptadas para llevar a cabo la combinación de clasificadores, a saber: *selección y fusión* (Valdovinos y col. 2005). Existen diferentes estudios relativos al tema de la combinación donde se destacan las ventajas de este planteamiento en el sentido de que la combinación permite resaltar las bondades de los clasificadores al mismo tiempo que se atenúan o desaparecen las desventajas (Partridge y Griffith, 2002; Deng y Zhang 2006).

En la selección, cada clasificador individual proporciona una decisión y se selecciona un único clasificador como el mejor de acuerdo a alguna estrategia preestablecida. Sin embargo, en la fusión los clasificadores se combinan utilizando diferentes estrategias (Kuncheva, 2004, Grim y col. 2002, Kittler y col. 1998 o Duda y Hart, 2000). Dentro de

estas estrategias se incluyen las denominadas: votación mayoritaria (“majority voting”), reglas de Máximo, Mínimo o de la Mediana.

La conducta de diferentes características ha sido estudiada en clasificación de texturas. Se entiende por característica alguna propiedad que describa el contenido de la textura (Valdovinos y col. 2005, Puig y García, 2006 o Hanmandlu y col. 2004). Existen dos orientaciones importantes en este aspecto, a saber: *a nivel de píxeles* y *a nivel de regiones*.

A nivel de píxeles se considera cada píxel de forma individual y se trata de clasificar el mismo como perteneciente a una determinada clase. En este caso las propiedades que se evalúan son sus componentes espectrales, esto es el color (Rud y col. 2006). En cuanto a los modelos de color utilizados destaca el modelo RGB, no obstante cualquiera de los modelos de color (HSV, YIQ, etc.) puede ser utilizado. En el trabajo de Yu y col. (2003) se analizan algunos de ellos. También en Kumar y col. (2001) se estudian algunos aspectos relacionados con el análisis multiespectral.

Los métodos a nivel de regiones no sólo tienen en cuenta cada píxel de forma individual sino los vecinos que le rodean. En este sentido lo que se evalúa son determinadas formas de aparición de las texturas siguiendo ciertos patrones. Existen diversas bases de imágenes con este tipo de texturas. Para ilustrar lo anterior citemos algunas de ellas: Brodatz, BME o VisTex por citar sólo algunas. En la Figura 2.1 se muestran algunas de tales texturas.



Figura 2.1 Diferentes ejemplos de texturas procedentes de las bases de imágenes Brodatz y BME

Para el análisis de las texturas mostradas en la figura 2.1 se utilizan diversos descriptores de vecindad o filtros (Wagner, 1999, Smith and Burns, 1997, Randen and Husoy, 1999, Pajares y Cruz, 2001, Escalera, 2001 o Gonzalez y Woods, 2001) un conjunto de ellos es el que se muestra a continuación:



- Matrices de co-ocurrencia con una serie de propiedades basadas en el histograma de las matrices tales como varianza, entropía, momentos de segundo orden, etc.
- Filtros de Gabor
- Máscaras de Laws
- Fourier

En la figura 2.2 se muestra una imagen representativa del conjunto de imágenes que se analizan en el presente trabajo. Tanto esta imagen como las utilizadas en esta investigación son cortesía del SITGA (Servicio Territorial de Galicia) y han sido proporcionadas por la empresa DIMAP (Digital Image Processing <http://www.dimap.es/>). Dichas imágenes constituyen el origen de la investigación que se presenta. Son imágenes aéreas correspondientes a la región de Abadín en Lugo (Galicia, España).

A la vista de la imagen mostrada se deduce que las texturas presentes en la misma (y por tanto en el conjunto de las analizadas) no se corresponden con el tipo de texturas mostrado en la Figura 2.1. En efecto, las diferencias a destacar son las siguientes:

- 1) En la figura 2.2 no aparecen patrones de textura repetitivos como en la figura 2.1, que permitan aplicar los descriptores que se usan para su tratamiento
- 2) No existe una única textura, puesto que se distinguen zonas boscosas, de praderas, zonas sin cultivar y construcciones relativas a núcleos de población.
- 3) Existe una clara distinción en las bandas espectrales (colores).



Figura 2.2 Ejemplo de imagen aérea con diferentes texturas naturales (cortesía del Servicio Territorial de Galicia en colaboración con Dimap)

Como consecuencia de las consideraciones expuestas anteriormente y aún siendo conscientes de que alguna de ellas ya se ha mencionado, en el presente trabajo de investigación se ha optado por lo siguiente:

- Utilizar una estrategia a *nivel de píxel*
- La descripción de cada píxel es en base a su color, considerando como propiedades las componentes RGB. Por tanto, un píxel está descrito por un vector  $\mathbf{x} \in \mathbb{R}^3$  cuyas componentes serán los valores R,G y B del mismo, es decir:  $\mathbf{x} \equiv (R,G,B)$ .
- Posponer para futuras investigaciones el análisis de otros modelos de color como IHS.
- Seleccionar tres clasificadores clásicos, que han sido ampliamente probados y verificados en la literatura aunque no se hayan utilizado de forma específica en el tratamiento de las texturas naturales.
- La utilización de un clasificador híbrido como una opción que puede mejorar los resultados respecto de los clasificadores individuales.

Con la utilización de un espacio tridimensional se evitan algunos problemas relacionados con el hecho de usar espacios de alta dimensionalidad, tal y como se explica en Ho (2002). Es decir, mayor número de componentes de  $\mathbf{x}$ .

## 2.2. Descripción de los clasificadores seleccionados

### 2.2.1. Agrupamiento borroso

El objetivo de la técnica de agrupamiento conocida como *Agrupamiento borroso o Fuzzy Clustering* consiste en dividir  $n$  objetos  $x \in X$  caracterizados por  $p$  propiedades en  $c$  clústeres o grupos. Supongamos el conjunto de datos  $X = \{x_1, x_2, \dots, x_N\} \in \mathbb{R}^p$  un subconjunto del espacio real  $p$ -dimensional  $\mathbb{R}^p$ . Cada  $x_k = \{x_{k_1}, x_{k_2}, \dots, x_{k_p}\} \in \mathbb{R}^p$  se denomina vector de características,  $x_{k_j}$  es la  $j$ -ésima característica de la observación  $x_k$ .

Este clasificador puede encontrarse perfectamente especificado en las siguientes referencias clásicas: Bezdek (1981), Duda y Hart, (2000) o Zimmerman (1991).

Puesto que los elementos de un clúster deben ser tan similares entre sí como sea posible y a la vez deben ser tan diferentes a los elementos de otros clústeres como también sea posible, el proceso se controla por el uso de medidas de similitud basadas en distancias. Así la similitud o la diferencia entre dos puntos  $x_k$  y  $x_l$  puede interpretarse como la distancia entre esos puntos.

Una distancia entre dos objetos de un universo  $X$  es una función  $d : X \times X \rightarrow R^+$  que toma valores reales que denotamos  $d(x_k, x_l) = d_{kl} \geq 0$  y que cumple tres propiedades:

1.  $d_{kl} = 0 \Leftrightarrow x_k = x_l$  para todo  $0 \leq k, l \leq N$
2. simetría:  $d_{kl} = d_{lk}$  para todo  $0 \leq k, l \leq N$
3. Desigualdad triangular  $d_{kl} \leq d_{kj} + d_{jl}$  para todo  $0 \leq k, j, l \leq N$  (2.1)

Cada partición del conjunto  $X = \{x_1, x_2, \dots, x_n\}$  puede enfocarse desde dos perspectivas: fuzzy y no fuzzy. Una partición no fuzzy se conoce en terminología inglesa como “crisp”. Si se desea realizar una partición del conjunto  $X$  en  $c$  clústeres tendremos  $S_i \{i = 1, \dots, c\}$  subconjuntos.

Para cada partición,  $S_i$  define un conjunto borroso  $\mu_i$  sobre el universo  $X$ ,  $\mu_i : X \rightarrow [0, 1]$  que asigna lo que se conoce como grado de pertenencia  $\mu_{ik}$  de cada objeto  $x_k$  al subconjunto  $S_i$ . Así pues denotamos dicho grado de pertenencia del elemento  $x_k$  al clúster  $S_i$  como  $\mu_i(x_k) = \mu_{ik}$ . En el caso de conjuntos “crisp” un objeto  $x_k$  se dice que pertenece a un  $S_i$  dado y no pertenece al resto. Esto se expresa con los valores discretos  $\{0, 1\}$  de la siguiente forma  $\mu_{ik} = 1$  para indicar pertenece y  $\mu_{ik} = 0$  para expresar que no pertenece. Por el contrario, en el caso de conjuntos fuzzy se dice que un objeto puede pertenecer a diferentes subconjuntos y así se habla por ejemplo de que  $x_k$  pertenece a un conjunto  $S_i$  con grado de pertenencia  $\mu_{ik}$  y a  $S_j$  con grado de pertenencia  $\mu_{jk}$ . Como ejemplo, supongamos que se tienen tres conjuntos  $S_i$ ,  $S_j$  y  $S_h$ , en este caso podríamos decir que el objeto  $x_k$  pertenece a los conjuntos con los siguientes grados de pertenencia  $\mu_{ik} = 0.4$ ,  $\mu_{jk} = 0.5$  y  $\mu_{hk} = 0.1$ . Los valores tomados pertenecen al intervalo continuo  $[0, 1]$ .

Dado  $X = \{x_1, x_2, \dots, x_n\}$  y el conjunto  $V_{cn}$  de todas las matrices reales de dimensión  $c \times n$ , con  $2 \leq c < n$ . Se puede obtener una matriz representando la partición de la siguiente manera  $U = \{\mu_{ik}\} \in V_{cn}$ . Tanto en el supuesto “crisp” como en el fuzzy se deben cumplir las siguientes condiciones:

- 1)  $\mu_{ik} \in \{0,1\}$  *crisp* o  $\mu_{ik} \in [0,1]$  *fuzzy*  $1 \leq i \leq c; 1 \leq k \leq n$
- 2)  $\sum_{i=1}^c \mu_{ik} = 1 \quad 1 \leq k \leq n$
- 3)  $0 < \sum_{k=1}^n \mu_{ik} < n \quad 1 \leq i \leq c$

(2.2)

Para ilustrar los conceptos anteriores sea  $X = \{x_1, x_2, x_3\}$  con él podríamos construir las siguientes particiones teniendo en cuenta que  $c = 2$

“*crisp*”

$$U = \begin{bmatrix} x_1 & x_2 & x_3 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

$$U = \begin{bmatrix} x_1 & x_2 & x_3 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

“*fuzzy*”

$$U = \begin{bmatrix} x_1 & x_2 & x_3 \\ 0.3 & 0.5 & 0 \\ 0.7 & 0.5 & 1 \end{bmatrix}$$

$$U = \begin{bmatrix} x_1 & x_2 & x_3 \\ 0.9 & 0.4 & 0.2 \\ 0.1 & 0.6 & 0.8 \end{bmatrix}$$

La localización de un clúster  $S_i$  se representa por su centro  $v_j = \{v_{j_1}, v_{j_2}, \dots, v_{j_p}\} \in \mathbb{R}^p$  con  $j = 1, \dots, c$ , alrededor del cual se concentran los objetos.

La definición básica de llevar a cabo el problema de la partición fuzzy para  $m > 1$  consiste en minimizar la función objetivo según la ecuación (2.3):

$$\min z_m(U; v) = \sum_{k=1}^n \sum_{j=1}^c \mu_{jk}^m \|x_k - v_j\|_G^2 \quad (2.3)$$

donde  $G$  es una matriz de dimensión  $p \times p$  que es simétrica y definida positiva. Así se puede definir una norma general del tipo,

$$\|x_k - v_j\|_G^2 = (x_k - v_j)^t G (x_k - v_j) \quad (2.4)$$

Diferenciando la función objetivo para  $v_i$  (suponiendo constante  $U$ ) y  $\mu_{jk}$  (suponiendo constante  $v$ ) y aplicando la condición de que  $\sum_{j=1}^c \mu_{jk} = 1$ , se obtiene,

$$v_j = \frac{1}{\sum_{k=1}^n (\mu_{jk})^m} \sum_{k=1}^n (\mu_{jk})^m x_k \quad j = 1, \dots, c \quad (2.5)$$

$$\mu_{jk} = \frac{\left( \frac{1}{\|x_k - v_j\|_G^2} \right)^{2/m-1}}{\sum_{h=1}^c \left( \frac{1}{\|x_k - v_h\|_G^2} \right)^{2/m-1}} \quad j = 1, \dots, c; k = 1, \dots, n \quad (2.6)$$

el exponente  $m$  se conoce como peso exponencial y disminuye la influencia del ruido al obtener los centros de los clústeres, reduciendo la influencia de los valores pequeños de  $\mu_{jk}$  (puntos lejos de  $v_j$ ) frente a valores altos de  $\mu_{jk}$  (puntos cercanos a  $v_j$ ). Cuanto mayor sea  $m > 1$  mayor es dicha influencia.

Existen diferentes medidas escalares para validar la partición, siendo el objetivo encontrar el número de clústeres que obtenga las mejores medidas. De entre todos ellos destaca por su importancia (Balasto y col. 2006, Kim y col. 2003) el *Coeficiente de partición (CP)* que mide la cantidad de solapamiento entre los clústeres

$$CP = \frac{1}{n} \sum_{j=1}^c \sum_{i=1}^n (\mu_{ji})^2 \quad (2.7)$$

Cuanto más próximo sea este valor a la unidad tanto mejor será la clasificación de los datos llegando a ser una partición pura cuando toma el valor de la unidad.

### 2.2.2. Clasificador Bayesiano

En el caso general y más típico de una distribución de probabilidad Gaussiana o Normal multivariable, ni la media  $\mathbf{m}$  ni la matriz de covarianza  $C$  son conocidas. Por tanto, esos parámetros desconocidos constituyen las componentes del vector de parámetros  $\mathbf{w} = \{\mathbf{m}, C\}$  consideremos el supuesto univariable con  $\mathbf{m} = m$  y  $C = \sigma^2$ , en cuyo caso

$$\ln p(x_i/\mathbf{w}) = -\frac{1}{2} \ln 2\pi C - \frac{1}{2C} (x_i - m)^2 \quad (2.8)$$

$$\nabla_{\mathbf{w}} \ln p(x_i/\mathbf{w}) = \begin{bmatrix} \frac{1}{C} (x_i - m) \\ -\frac{1}{2C} + \frac{(x_i - m)^2}{2C^2} \end{bmatrix} \quad (2.9)$$

La minimización sobre los datos de entrenamiento conduce ahora a las condiciones,

$$\frac{1}{n} \sum_{i=1}^n \frac{1}{\hat{C}} (x_i - \hat{m}) = 0 \quad -\frac{1}{n} \sum_{i=1}^n \frac{1}{2\hat{C}} + \frac{1}{n} \sum_{i=1}^n \frac{(x_i - \hat{m})^2}{2\hat{C}^2} = 0 \quad (2.10)$$

donde  $\hat{m}$  y  $\hat{C}$  son las estimas de máxima verosimilitud para  $m$  y  $C$ , respectivamente.

Sustituyendo  $\hat{m}$  y  $\hat{\sigma}^2 = \hat{C}$  obtenemos las estimas de máxima verosimilitud para  $m$  y  $\sigma^2$

$$\hat{m} = \frac{1}{n} \sum_{i=1}^n x_i \quad \hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^n (x_i - m)^2 \quad (2.11)$$

Aunque el análisis del caso multivariable es básicamente muy similar, se requiere mucha más manipulación. El resultado muy bien conocido en estadística es que las estimas de máxima verosimilitud para  $\mathbf{m}$  y  $C$  están dadas por,

$$\mathbf{m} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \quad C = \frac{1}{n-1} \sum_{i=1}^n (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^t \quad (2.12)$$

La expresión (2.12) nos dice que la estima de máxima verosimilitud para el vector media es la media simple. La estima de máxima verosimilitud para la matriz de covarianza es la media aritmética de las  $n$  matrices  $(\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^t$ . Puesto que la verdadera matriz de covarianza es el valor esperado de la matriz  $(\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^t$ , se obtiene un resultado muy satisfactorio.

Una vez estimados los parámetros  $\mathbf{m}$  y  $C$ , la función de densidad de probabilidad queda perfectamente especificada por la ecuación dada en (2.13) suponiendo que dicha función sigue una distribución Gaussiana.

$$p(\mathbf{x} | \mathbf{m}, C) = \frac{1}{(2\pi)^{d/2} |C|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \mathbf{m})^t C^{-1} (\mathbf{x} - \mathbf{m}) \right\} \quad (2.13)$$

Según la teoría general de la probabilidad de Bayes, dado  $\mathbf{x}$  el objetivo que se plantea es asignar  $\mathbf{x}$  alguna de las clases existentes. Supongamos que las clases son  $w_1, w_2, \dots, w_c$  para cada una de ellas se puede estimar la función de densidad de probabilidad dada en (2.13)

obteniendo  $(\mathbf{m}_1, C_1)$  para la clase  $w_1$ ,  $(\mathbf{m}_2, C_2)$  para la clase  $w_2$  y así sucesivamente hasta llegar a la clase  $w_c$  con  $(\mathbf{m}_c, C_c)$ .

Por tanto, dada la observación  $\mathbf{x}$  se trata de determinar la probabilidad a posteriori de que dicha observación pertenezca a la clase  $w_j$ . El proceso para llevar a cabo esta operación se puede realizar por medio de la teoría de la probabilidad de Bayes calculando la probabilidad a posteriori como sigue,

$$P(w_j | \mathbf{x}) = \frac{p(\mathbf{x} | \mathbf{m}_j, C_j)P(w_j)}{\sum_{j=1}^c p(\mathbf{x} | \mathbf{m}_j, C_j)} \quad (2.14)$$

El numerador del segundo miembro de la ecuación (2.14) incluye dos términos, a saber: 1) la función de densidad de probabilidad  $p(\mathbf{x} | \mathbf{m}_j, C_j)$  que está dada en la ecuación (2.13) y 2) la probabilidad a priori  $P(w_j)$  de que dicha muestra pertenezca a la clase  $w_j$ . Uno de los problemas que se plantean respecto del cálculo de la probabilidad a priori consiste en cómo determinar esta probabilidad antes de observar la muestra.

En cualquier caso, la decisión final sobre la asignación de  $\mathbf{x}$  a una clase dada se basa en el siguiente criterio, según Duda y col. (2000):

$$\mathbf{x} \in w_j \quad \text{si} \quad P(w_j | \mathbf{x}) > P(w_k | \mathbf{x}) \quad \forall k \neq j \quad (2.15)$$

Teniendo en cuenta la ecuación (2.14) este criterio se puede expresar como sigue ya que el denominador en (2.14) representa la densidad de probabilidad mixta, que no tiene carácter discriminante puesto que es la misma en todos los casos.

$$\mathbf{x} \in w_j \quad \text{si} \quad p(\mathbf{x} | w_j)P(w_j) > p(\mathbf{x} | w_k)P(w_k) \quad \forall k \neq j \quad (2.16)$$

Como se ha mencionado anteriormente, la única cuestión pendiente estriba en el cómputo de la probabilidad a priori. Cuando ésta no se conoce, lo que se hace es fijarla para todas las clases a un valor constante, por ejemplo a 0.5 para que no intervenga realmente en la decisión.

### 2.2.3. Ventana de Parzen

A diferencia de la estimación paramétrica, donde la función de densidad de probabilidad  $p(\mathbf{x})$  se obtiene estimando los parámetros desconocidos de un modelo conocido, en la estimación no paramétrica no se conoce el modelo, es decir no se sabe si sigue una distribución Gaussiana o cualquier otra (Parzen, 1962, Duda y col. 2000). En este caso la estima se lleva a cabo considerando todas las muestras disponibles para cada clase, tal y como se especifica a continuación,

$$p(\mathbf{x}) = \frac{1}{n} \sum_{j=1}^n \left\{ \frac{1}{(2\pi)^{p/2} |C|^{1/2} h^p} \exp \left[ -\frac{1}{2h^2} (\mathbf{x} - \mathbf{x}_j)^T C^{-1} (\mathbf{x} - \mathbf{x}_j) \right] \right\} \quad (2.17)$$

## 2.3. Métodos clásicos de combinación de clasificadores

El tema relativo a la combinación de clasificadores o también denominada hibridación ha sido ampliamente estudiado en la literatura, por ejemplo Partridge y Griffith (2002) establecen un marco de trabajo, fijando los criterios fundamentales para diseñar un método híbrido que utiliza diferentes clasificadores.

Existen algunas alternativas sobre la combinación de clasificadores ampliamente difundidas en la literatura especializada sobre el tema. Siguiendo las referencias Kuncheva (2004), Kittler y col. (1998) o Duda y col. (2000) el problema se plantea en los términos que se describen a continuación.

Dada una muestra para ser clasificada,  $\mathbf{x}_s$  ésta se suministra a los distintos clasificadores, para su procesamiento. Supongamos que disponemos de  $m$  clasificadores y varias clases  $c$  donde la muestra puede ser asignada. En la literatura se han descrito tres métodos de combinación con una aceptación generalizada.

En Kittler y col. (1998) se establece un marco de referencia para la combinación de clasificadores desde el punto de vista probabilista, es decir que las salidas proporcionadas por cada clasificador son precisamente de naturaleza probabilista. Se trata de un planteamiento muy próximo al que hacemos en el presente trabajo de investigación. Por tanto los métodos de combinación allí expresados constituyen buenas referencias para ser



utilizadas como medidas de comparación con respecto a la propuesta de combinación realizada en este trabajo. En concreto son tres los métodos expresados en forma de reglas, a saber:

- Reglas de máximo
- Reglas de mínimo
- Regla de media

Existe también un método muy utilizado denominado *votador mayoritario* (“*majority voting*”), donde cada clasificador determina la clase a la que pertenece una muestra dada y a continuación dependiendo del número de votos obtenido por cada clase se determina su pertenencia.

En la tabla 2.1 se especifica cada una de ellas.

	Clase 1	Clase 2		Clase $c$	
<b>Clasificador 1</b>	$^1S_1$	$^1S_2$	...	$^1S_c$	
<b>Clasificador 2</b>	$^2S_1$	$^2S_2$	...	$^2S_c$	
<b>Clasificador <math>m</math></b>	$^mS_1$	$^mS_2$	...	$^mS_c$	
<i>Máximo</i>	$^hS_1$	$^hS_2$	...	$^hS_c$	<i>Regla máximo:</i> $^H S_J = \max_{j=1}^c \max_{h=1}^m \{^h S_j\}$
					<i>Regla mínimo:</i> $^H S_J = \min_{j=1}^c \max_{h=1}^m \{^h S_j\}$
<i>Media</i>	$^M S_1 = \frac{1}{m} \sum_{i=1}^m {}^i S_1$	$^M S_2 = \frac{1}{m} \sum_{i=1}^m {}^i S_2$		$^M S_c = \frac{1}{m} \sum_{i=1}^m {}^i S_c$	<i>Regla media</i> $^H S_J = \max_{j=1}^c \{^M S_j\}$

Tabla 2.1 Métodos sobre combinación de clasificadores

## 2.4. Combinación mediante operadores de agregación

Una de las propuestas novedosas realizada en este trabajo consiste precisamente en la utilización de los denominados operadores de agregación fuzzy.

En la sección siguiente 2.4.1 se exponen los aspectos generales teóricos en los que se fundamentan los denominados operadores de agregación mientras que en la sección 2.4.2 se incluyen diversos tipos de operadores de esta naturaleza.

### 2.4.1. Aspectos generales de la teoría de agregación fuzzy

La agregación de información de manera eficiente y flexible se ha convertido en la principal tarea de los problemas de acceso de información y otros problemas de decisión multicriterio, puesto que precisan procesar una gran cantidad de información cuya calidad y precisión es muy variada. En particular, resultan muy útiles las agregaciones entre el operador mínimo y el operador máximo a través de los operadores de medias. En el desarrollo de este clasificador híbrido se comprueban y contrastan resultados de algunos de los operadores más relevantes derivados de las diferentes lógicas, que se utilizan para su definición.

Evidentemente no existe un único criterio para seleccionar los operadores de agregación, y esto ha hecho que se propongan algunas condiciones que se deben tener en cuenta para elegirlos, algunas de las cuales se citan en Zimmermann (1991), tales como las siguientes:

1. *Fuerza axiomática.* En igualdad de condiciones, un operador es mejor cuanto menos limitado esté por los axiomas que satisface.
2. *Ajuste empírico.* Además de que los operadores satisfagan ciertos axiomas o tengan ciertas cualidades formales, deben reflejar adecuadamente la realidad, lo que normalmente sólo puede ser testado con pruebas empíricas.
3. *Adaptabilidad.* Los operadores tienen que adaptarse al contexto específico en el que se encuentran, esencialmente, mediante la parametrización. Por ejemplo, los operadores mín y

máx no son nada flexibles, mientras que los operadores OWA pueden ser adaptados a ciertos contextos eligiendo parámetros adecuados.

4. *Eficiencia numérica*. El esfuerzo computacional de cálculo es especialmente importante cuando se tienen que resolver problemas grandes. De hecho, en muchas ocasiones debe recurrirse a técnicas heurísticas capaces de encontrar soluciones de calidad aunque no sean necesariamente óptimas (Herrera y col., 1996).

5. *Compensación y rango de compensación*. Cuanto mayor sea el grado en que se contrarrestan las funciones de pertenencia de los conjuntos borrosos agregados, el operador de agregación representará mejor las situaciones en las que unos atributos son compensados por otros.

6. *Comportamiento agregado*. El grado de pertenencia de un conjunto borroso en el conjunto agregado depende muy frecuentemente del número de conjuntos combinados. De este modo, cada conjunto adicional añadido normalmente disminuirá los grados agregados de pertenencia resultantes, y este hecho debe ser tenido en cuenta cuando se elige el operador.

7. Nivel de escala requerido de las funciones de pertenencia. Diferentes operadores pueden requerir diferentes niveles de escala (nominal, intervalo, ratio o absoluto) de información de pertenencia para ser admisibles. En igualdad de condiciones, se prefiere el operador que requiere el nivel de escala más bajo.

### **2.4.2. Tipos de operadores de agregación**

Para la definición de los operadores fuzzy se recurre a la definición de las denominadas *t-normas* y *t-conormas*.

Las t-normas se comportan como conjunciones por lo que son ampliamente utilizadas en lógica borrosa. Las utilizamos por ello en la definición de expresiones de las medidas de especificidad. Además las t-normas pueden ser utilizadas como generadoras de *modus ponens*.

Una norma triangular (o brevemente una t-norma) es una operación binaria asociativa en  $[0, 1]$ ,  $T: [0, 1] \times [0, 1] \rightarrow [0, 1]$ , que para todo  $x, y, z$  de  $[0, 1]$  satisface:

- |                                                                 |                     |
|-----------------------------------------------------------------|---------------------|
| T1) $T(x, 1) = x$ ,                                             | (elemento neutro 1) |
| T2) Si $x \geq x', y \geq y'$ entonces $T(x, y) \geq T(x', y')$ | (monotonía)         |
| T3) $T(x, y) = T(y, x)$                                         | (simetría)          |
| T4) $T(x, T(y, z)) = T(T(x, y), z)$                             | (asociatividad)     |

lo que significa que es asociativa, conmutativa, no decreciente y con una condición de contorno. Puede visualizarse como una superficie sobre el cuadrado unidad que contiene al segmento  $(0,1,0)$   $(0,0,0)$ , al segmento  $(0,0,0)$   $(1,0,0)$  y al punto  $(1,1,1)$ . Esta definición puede generalizarse a conjuntos n-arios utilizando la propiedad asociativa. Las t-normas son muy utilizados en lógica borrosa para definir la intersección entre conjuntos borrosos, ya que generaliza la intersección clásica. Como operador lógico, son operadores que satisfacen la tabla lógica del conectivo lógico “y” (AND).

La mayor t-norma es el mínimo: Mín, que es continua. . Otras muy utilizadas son las de la familia del producto y las t-normas de la familia de Łukasiewicz (W). Verifican que  $\text{Min}(x, y) \geq \text{Prod}(x, y) \geq W(x, y)$ .

Las t-conormas, o conormas triangulares, son operadores  $S: [0,1] \times [0,1] \rightarrow [0,1]$  muy utilizados en lógica borrosa para definir las uniones entre conjuntos borrosos pues generalizan la unión clásica. Como operador lógico, son operadores que satisfacen la tabla lógica del conectivo “o”, (OR).

Se pueden definir las t-conormas a partir de las t-normas, axiomáticamente o mediante funciones generadoras (Schweizer y Sklar, 1983).

Una operación  $S: [0, 1] \times [0, 1] \rightarrow [0, 1]$  es una t-conorma si  $T(x, y) = 1 - S(1-x, 1-y)$  es una t-norma.

Dada una t-norma  $T$ , se define la conorma dual de  $T$  como  $T^*(x, y) = 1 - T(1-x, 1-y)$ .

La t-conorma dual del Mínimo es el Máximo, la dual del producto es la suma probabilística:  $\text{Prod}^*(x, y) = x + y - xy$ . La t-conorma dual de Łukasiewicz es  $W^*(x, y) = \text{Mín}\{1, x+y\}$ .

Axiomáticamente se define una conorma triangular  $S: [0, 1] \times [0, 1] \rightarrow [0,1]$ , como un operador que satisface los cuatro axiomas siguientes.

- S1)  $S(x, 0) = S(0, x) = x$ , para todo  $x \in [0,1]$
- S2) Si  $x \geq x', y \geq y'$  entonces  $S(x, y) \geq S(x', y')$  (monotonía)
- S3)  $S(x, y) = S(y, x)$  para todo  $x, y \in [0,1]$  (simetría)
- S4)  $S(x, S(y, z)) = S(S(x, y), z)$  para todo  $x, y, z \in [0,1]$  (asociatividad)

Para las definiciones de las t-normas y t-conormas existen diferentes lógicas, entre ellas destacan la lógica de Zadeh, la del producto y la de Łukasiewicz. De las lógicas anteriormente mencionadas derivan los operadores que se exponen a continuación. En la tabla 2.2 se muestran una serie de operadores de agregación en los que se combinan dos medidas fuzzy  $s, c \in [0,1]$  que se combinan mediante las operaciones indicadas para generar una nueva medida fuzzy. Algunos de tales operadores están parametrizados, de forma que se indica el tipo de parámetros que requiere en su caso junto con el rango de variabilidad de los mismos.

Tabla 2.2 Lista de operadores de **Operadores de Agregación Fuzzy (OAF)**

Lógica	Intersección (AND) t-noma	Unión (OR) t-conorma	Negación
Zadeh	$\text{Min}(x,y)$	$\text{Max}(x,y)$	$1-x$
	$x \cdot y$	$x + y - xy$	
$Yager_p$	$1 - \text{Min}(((1-x)^p + (1-y)^p)^{1/p}, 1)$	$\text{Min}((x^p + y^p)^{1/p}, 1)$	$(1-x^p)^{1/p}$
$Dombi_\lambda$ $\lambda > 0$	$\left(1 + \left[\left(\frac{1}{x} - 1\right)^{-\lambda} + \left(\frac{1}{y} - 1\right)^{-\lambda}\right]^{-1/\lambda}\right)^{-1}$	$\left(1 + \left[\left(\frac{1}{x} - 1\right)^\lambda + \left(\frac{1}{y} - 1\right)^\lambda\right]^{1/\lambda}\right)$	$1-x$
$Weber_\lambda$ $\lambda > -1$	$\text{Max}\left(\frac{x + y - 1 + \lambda xy}{1 + \lambda}, 0\right)$	$\text{Min}(x + y + \lambda xy, 1)$	$\frac{1-x}{1+\lambda x}$
Łukasiewicz: $Weber_\lambda \lambda = 0$	$\text{Max}(x+y-1, 0)$	$\text{Mín}\{1, x+y\}$	$1-x$
$Hamacher_\gamma$ $\gamma > 0$	$\frac{xy}{\gamma + (1-\gamma)(x+y-xy)}$	$\frac{x+y-xy-(1-\gamma)x}{1-(1-\gamma)xy}$	$1-x$

Notese que los operadores de la lógica del producto son los de Hamacher con el parámetro  $\gamma = 1$ .

Teniendo en cuenta los valores obtenidos en la tabla 2.1, donde para una muestra  $x_s$ , cada clasificador proporciona un valor de pertenencia a cada una de las clases. Así pues, el objetivo consiste en combinar (agregar desde el punto de vista fuzzy) los valores proporcionados por cada clasificador.

El proceso se lleva a cabo teniendo en cuenta las columnas y filas de la tabla 2.1. Básicamente consiste en agregar los dos primeros valores de cada columna, el resultado se agrega con el tercer valor de la columna y así sucesivamente hasta agregar el último valor de la última columna. El proceso se sintetiza teniendo en cuenta que las variables  $x$  e  $y$  de la tabla 2.2 se corresponden con las  ${}^iS_j$  de la tabla 2.1.

1. Dada una clase  $j$  y por tanto una columna de dicha tabla agregar mediante el operador  $OAF$  los dos primeros valores de la columna:

$${}^1A_j = OAF({}^1S_j, {}^2S_j) \quad (2.18)$$

2. Desde  $h = 1$  hasta  $m$  realizar la agregación como sigue:

$${}^hA_j = OAF({}^{h-1}A_j, {}^hS_j) \quad (2.19)$$

Al finalizar la agregación de los valores dos a dos se obtiene para cada clase un valor resultante agregado que resulta ser  ${}^hA_j$ .

Según lo anterior y como resultado de la agregación para cada una de las clases tenemos finalmente ese valor resultante agregado  ${}^hA_j$ . Se hace necesario seleccionar de entre todos ellos el más conveniente según algún criterio especificado. En el capítulo siguiente, este criterio se establece como el máximo en  $j$  de entre todos los  ${}^hA_j$  disponibles por las razones allí expresadas.

Esta agregación hecha mediante operadores  $OAF$  de t-normas y t-conormas generaliza las operaciones de intersección y unión de conjuntos clásicos. En el caso clásico, si  ${}^mS_c$  es un conjunto que indica si un píxel pertenece a la clase  $c$  según el método  $m$ , podemos representarlo mediante su función característica:

${}^mS_c(x_j) = 1$  si  $x_j$  pertenece al clúster  $c$  según el método  $m$

${}^mS_c(x_j) = 0$  si  $x_j$  no pertenece al clúster  $c$  según el método  $m$

Si quisiésemos hacer la unión (intersección) de dos conjuntos, se hace mediante el operador lógico OR (AND), y su función característica queda de la siguiente forma:

$$({}^1S_c \cup {}^2S_c)(x_j) = {}^1S_c(x_j) \vee {}^2S_c(x_j)$$

$$({}^1S_c \cap {}^2S_c)(x_j) = {}^1S_c(x_j) \wedge {}^2S_c(x_j)$$

En el caso de métodos borrosos (denotamos  $\mu$  al conjunto borroso, en lugar de  $S$ ), se pueden generalizar las uniones e intersecciones con t-normas  $T$  y t-conormas  $S$  de la siguiente manera:

$$({}^1\mu_c \cup {}^2\mu_c)(x_j) = T({}^1\mu_c(x_j), {}^2\mu_c(x_j))$$

$$({}^1\mu_c \cap {}^2\mu_c)(x_j) = S({}^1\mu_c(x_j), {}^2\mu_c(x_j))$$

Notese que en el caso de valores nítidos (0 ó 1), todas las t-normas verifican la tabla de verdad del operador lógico AND ( $\wedge$ ) y todas las t-conormas generalizan el OR ( $\vee$ ), por lo que estamos generalizando las operaciones algebraicas clásicas.

Visto desde este punto de vista, entendemos la agregación dada en (2.19) como

${}^mA_c = {}^1S_c \cup \dots \cup {}^mS_c$  o bien como  ${}^mA_c = {}^1S_c \cap \dots \cap {}^mS_c$  según que el operador de agregación OAF sea elegido como una t-conorma o como una t-norma.

Así pues, el método del máximo y del mínimo dado en la Tabla 2.1 sería un caso particular de hibridación usando la unión e intersección de métodos con operadores de la lógica de Zadeh. En el caso de la unión, un píxel pertenece a un clúster del método híbrido si pertenece dicho clúster en alguno de los métodos, y en el caso de la intersección, se generaliza el concepto de que, un píxel pertenece a un clúster del método híbrido si pertenece a dicho clúster en todos los métodos.





## Capítulo 3

### 3. Diseño de un clasificador híbrido no supervisado

#### 3.1. Introducción

Antes de comenzar con el proceso de diseño de un clasificador híbrido conviene destacar las fases o mecanismos involucrados en todo proceso de aprendizaje-clasificación. La figura 3.1 muestra un esquema general donde se detalla esto.

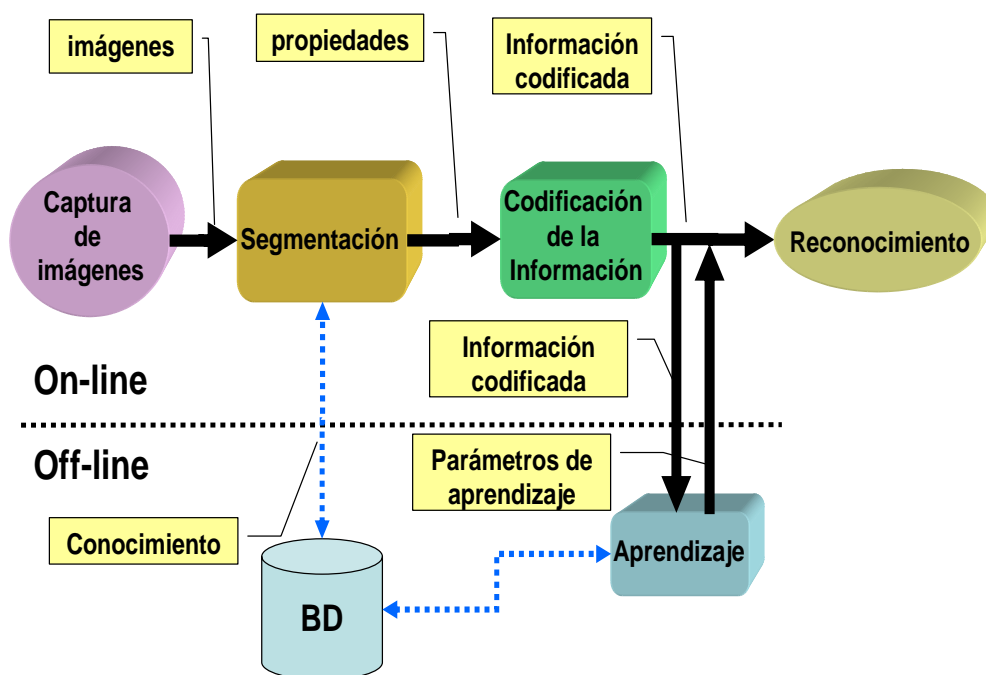


Figura 3.1 Esquema de un proceso aprendizaje-clasificación

Existen dos procesos perfectamente identificados, a saber: *on-line* y *off-line*. Durante el proceso *on-line* se lleva a cabo el aprendizaje propiamente dicho, mientras que en el *off-line* se realiza la clasificación o reconocimiento. Los pasos a seguir en ambos procesos son prácticamente similares salvo en la parte de aprendizaje que es exclusivo del proceso *off-line*.

Los pasos a seguir durante el proceso *off-line* son los siguientes:

**Captura de imágenes:** se obtienen las imágenes a partir de una fuente de imágenes existentes o bien a partir de la captura directa de la escena mediante los correspondientes dispositivos de captura.

**Segmentación:** las imágenes son procesadas, extrayendo de ellas los elementos o estructuras. En el caso que nos ocupa se trata de extraer los píxeles, que constituyen las *propiedades* de las imágenes. Si existen propiedades previamente extraídas y almacenadas en la Base de Datos (BD) o Base de Conocimiento (BC), éstas se recuperan para añadirlas a las que se acaban de obtener. Todas ellas forman el conjunto de muestras de entrenamiento. Posteriormente, serán almacenadas todas ellas de nuevo en la BC para un posterior proceso de aprendizaje.

**Codificación de la información:** la información se dispone de forma que pueda ser procesada adecuadamente, normalmente será codificada en forma de vector o matriz y además normalizada en unos rangos de variabilidad apropiados.

**Aprendizaje:** con esta información se aplican los métodos de aprendizaje que se deseen, tras los cuales se *aprenden*, es decir se obtienen los parámetros correspondientes según el método aplicado. Estos parámetros se almacenan en la BC con el fin de poderlos recuperar durante la fase de clasificación *on-line*. Los parámetros aprendidos dependen del tipo de algoritmo aplicado según se expresa a continuación:

**Fuzzy Clustering:** centros de los clústeres y grados de pertenencia a los mismos

**Clasificador Bayesiano:** centros de los clústeres y matriz de covarianza de los mismos

**Clasificador de Parzen:** matrices de covarianza y muestras de las clases

Los pasos a seguir durante el proceso *on-line* son los siguientes:

Ahora se reciben imágenes nuevas que deben ser clasificadas, para ello se aplican los pasos de captura de imágenes, segmentación y codificación de la información de la misma manera que en el caso *off-line*. La diferencia estriba en el módulo de **Reconocimiento**, el cual tras recibir los datos codificados y haciendo uso de los parámetros de aprendizaje

almacenados en la BD procede a clasificar las texturas identificadas en las imágenes de entrada.

Las muestras clasificadas durante este proceso pueden incorporarse a la BD para llevar a cabo un nuevo proceso de *aprendizaje* de forma que los parámetros de aprendizaje puedan actualizarse con un mayor número de muestras de entrenamiento.

## 3.2. Antecedentes

La motivación para el diseño del clasificador híbrido que se propone en el presente trabajo tiene sus antecedentes en un primer intento por aplicar el clasificador de Bayes en toda su extensión es decir, teniendo en cuenta que la decisión de clasificación mediante Bayes tiene en cuenta tanto la función de densidad de probabilidad estimada según la ecuación (2.13) como la probabilidad a priori, ecuaciones (2.14) a (2.16).

Como ya se indicó, el principal problema a la hora de aplicar Bayes surge por el hecho de que la probabilidad a priori resulta desconocida. Para resolver este asunto, se propone el esquema que se muestra en la figura 3.2. El planteamiento que se hace en este planteamiento es el siguiente:

### *Fase de entrenamiento*

1. Dadas las muestras iniciales, que constituyen las entradas al sistema, con ellas se realiza una estima de la función de densidad de probabilidad de Bayes así como la obtención de los centros de los clústeres mediante el clasificador fuzzy.
2. Tanto los parámetros estimados en Bayes ( $\mathbf{m}_j$ ,  $C_j$ ) como los centros fuzzy ( $\mathbf{v}_j$ ) se almacenan en la base de conocimiento *BC*.

### *Fase de clasificación*

3. Ante la llegada de cada nueva muestra  $\mathbf{x}_s$ , se trata de decidir a qué clase pertenece. Para ello se recuperan de la *BC* el conocimiento almacenado previamente durante la fase de entrenamiento:  $\mathbf{m}_j$ ,  $C_j$ ,  $\mathbf{v}_j$ .
4. Para cada  $\mathbf{v}_j$  se obtiene el grado de pertenencia de dicha muestra a cada una de las clases según el procedimiento descrito en el capítulo previo, a saber  $\mu_{sj}$ . Estos valores constituyen un conocimiento previo sobre la pertenencia de la muestra a las

clases, en base a este hecho se considera que dicho grado de pertenencia puede actuar como la probabilidad a priori para el clasificador Bayesiano. Por tanto, la probabilidad a priori de que la muestra  $\mathbf{x}_s$  pertenezca a la clase  $w_j$  se obtiene como sigue  $P(w_j) = \mu_{sj}$ .

5. Conocida la probabilidad a priori y calculando la verosimilitud a través de la función de densidad de probabilidad, la probabilidad a posteriori se obtiene fácilmente a través de la ecuación (2.16).
6. La decisión final se toma en función de que la muestra pertenecerá a la clase que haya generado el mayor valor de la probabilidad a posteriori.

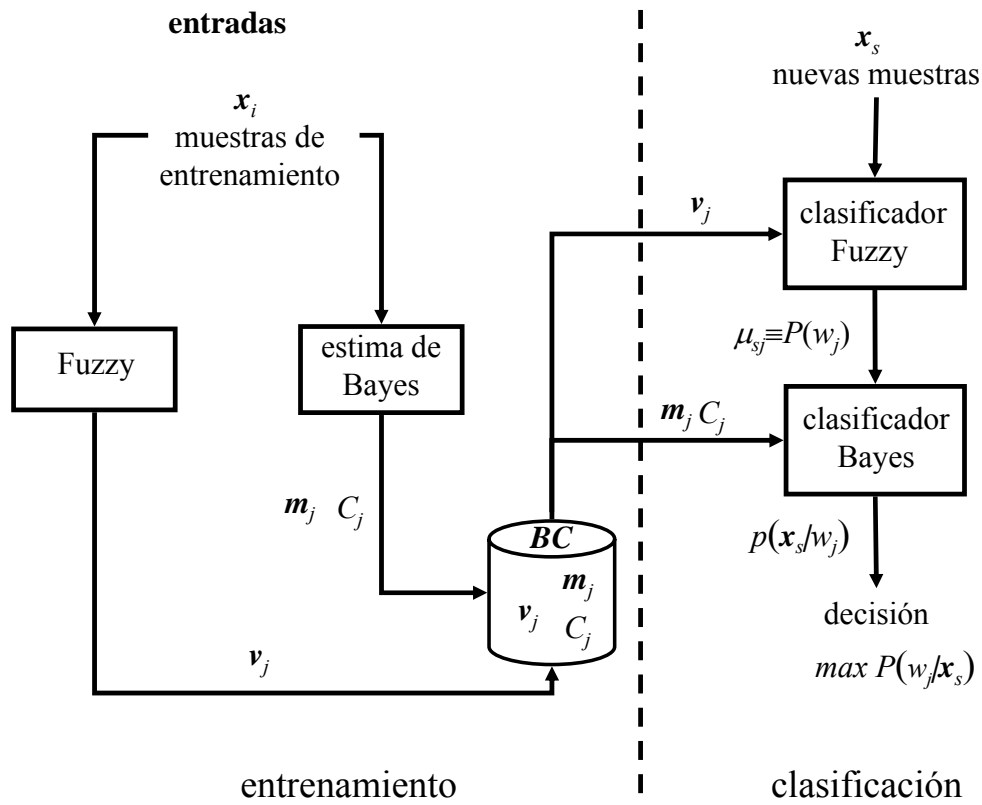


Figura 3.2 Clasificador de Bayes con el clasificador fuzzy que proporciona la probabilidad a priori

A partir de la descripción previa, se deduce que en realidad lo que se hace es un producto del resultado proporcionado por el clasificador fuzzy con el proporcionado con el clasificador de Bayes. Este esquema resulta ser precisamente uno de los métodos de agregación utilizados en la lógica fuzzy, concretamente mediante la *t-norma* de la lógica del producto, tal y como se ha descrito previamente. De este hecho surge la propuesta que se lleva a cabo en la siguiente sección.

### 3.3. Diseño del método híbrido

En la figura 3.3 se muestra el esquema general del procedimiento híbrido propuesto como solución al problema de la clasificación de texturas naturales. En él se distinguen las dos fases ya mencionadas y perfectamente definidas, a saber: entrenamiento y clasificación.

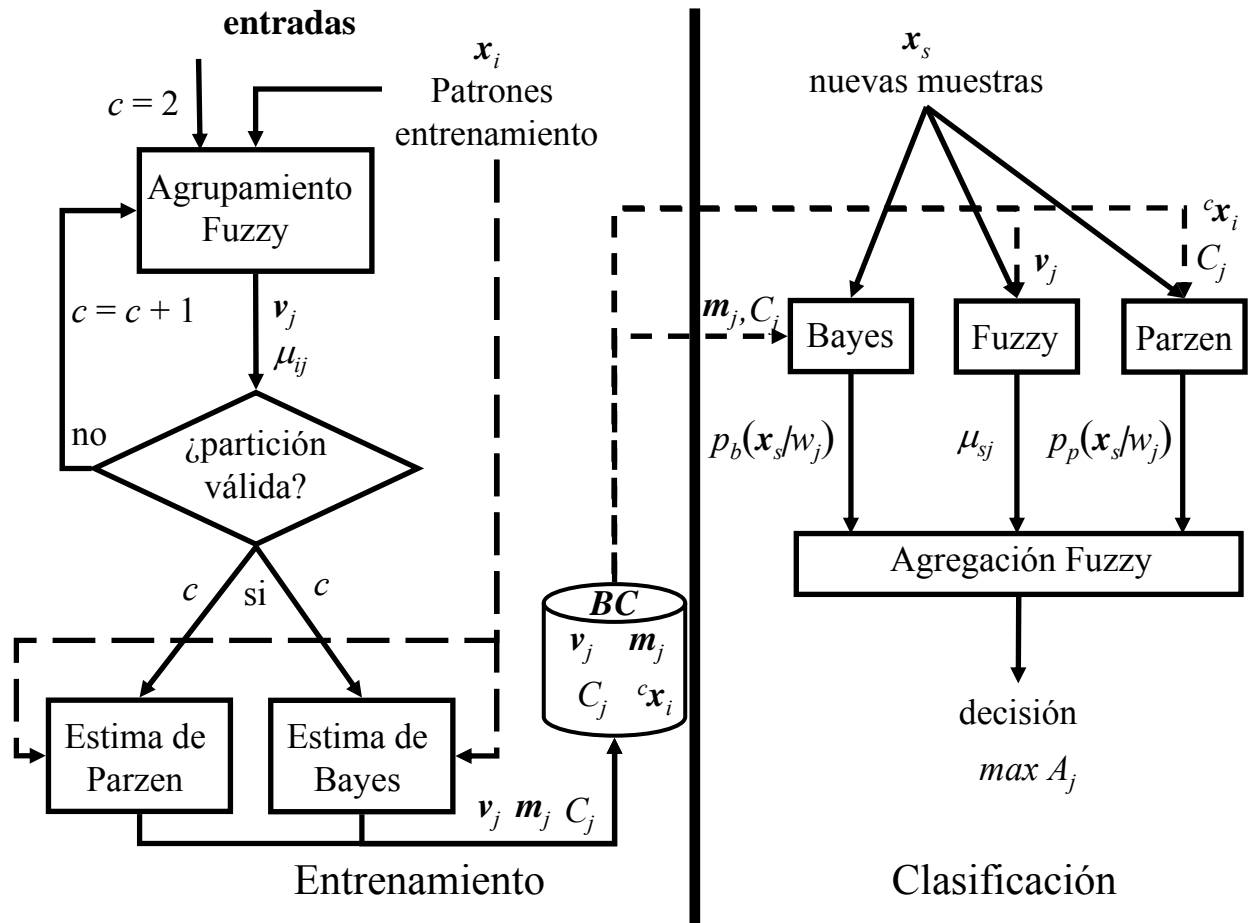


Figura 3.3 Diseño del clasificador híbrido

#### 3.3.1. Fase de entrenamiento

Se dispone de un conjunto de  $N$  muestras o patrones de entrenamiento  $X = \{x_1, x_2, \dots, x_N\}$ , del que inicialmente no se conoce su distribución en clases. Ni siquiera el número de clases. Por consiguiente la primera tarea del método propuesto consiste en realizar una primera distribución de las muestras en clases. Esto se lleva a cabo de la siguiente manera,

##### a) Agrupamiento Fuzzy: partición inicial

- 1) Fijar el número de clases  $c = 2$

- 2) Realizar una primera partición mediante el procedimiento descrito en Balasko y col. (2006) proporcionado a continuación

$$\text{Normalización: } Y = \frac{X - \mathbf{MIN}}{\mathbf{MAX} - \mathbf{MIN}}; \mathbf{MIN} = \min\{X\}; \mathbf{MAX} = \max\{X\} \quad (3.1)$$

Como no nos han dado los valores iniciales de los centros de los clústeres, se utiliza el procedimiento pseudoaleatorio descrito también en la misma referencia, dado por la siguiente expresión y teniendo en cuenta que el número de clústeres es dos.

$$\mathbf{v} = cD\overline{\mathbf{M}} \circ \mathbf{R} + D\overline{\mathbf{m}} \quad (3.2)$$

donde  $\overline{\mathbf{m}}$  es la media de los valores de  $Y$  con dimensión  $1 \times p$ ,

$\overline{\mathbf{M}} = \max(\text{abs}(Y - \overline{\mathbf{m}}))$  es una matriz de dimensión  $1 \times p$ ,  $D = \begin{bmatrix} \overbrace{1 \cdots 1}^c \\ \vdots \\ 1 \end{bmatrix}^t$  cuya

dimensión es  $c \times 1$ ;  $\mathbf{R}$  es una matriz de números aleatorios de dimensión  $c \times p$  donde cada elemento de la matriz se obtiene utilizando la función  $(\text{rand}(\cdot) - 0.5)$ , los valores aleatorios están restringidos al rango  $[0,1]$ . El operador  $\circ$  expresa multiplicación de matrices elemento a elemento.

- 3) Con los centros de los clústeres así obtenidos, se determinan los grados de pertenencia de cada una de las muestras a los clústeres representados por dichos centros, es decir se obtienen los grados de pertenencia  $\mu_{ij}$  a través de las ecuaciones (2.5) y (2.6).
- 4) Con los grados de pertenencia así obtenidos, se procede a **validar** la partición. Para ello se obtiene el valor del coeficiente de partición ( $CP$ ) mediante la ecuación (2.7). Como se explica allí este coeficiente varía entre 0 y 1. Una perfecta partición según este coeficiente se obtiene para un valor 1 y una partición muy mala para un valor de cero. En nuestro caso consideramos que la partición es válida si el valor de dicho coeficiente supera un valor de umbral  $T$ , fijado a 0.8 en el presente trabajo.
- 5) En el supuesto de que dicho umbral no sea superado por el coeficiente  $CP$ , se procede a buscar una nueva partición válida, para lo cual se incrementa el número de clústeres en una unidad haciendo  $c = c + 1$ , repitiendo íntegramente los pasos 2)

a 4). El proceso se repite hasta que se consigue validar la partición. En este momento los datos iniciales contenidos en el conjunto  $X$  se distribuyen de acuerdo al número de clases obtenidas. Para asignar cada muestra  $x_i$  a una clase  $w_j$  se determina cuál es el máximo valor de los grados de pertenencia de  $x_i$  a las diferentes clases, asignando  $x_i$  a la clase que genere el mayor grado de pertenencia. Tras la validación de la partición se obtiene una distribución de patrones por clases tal y como se muestra en la figura 3.4. Si bien, con los centros desnormalizados para que sus valores estén en el rango de los originales. Esta desnormalización se lleva a cabo mediante la siguiente expresión,

$$v = v (MAX - MIN) + MIN \quad (3.3)$$

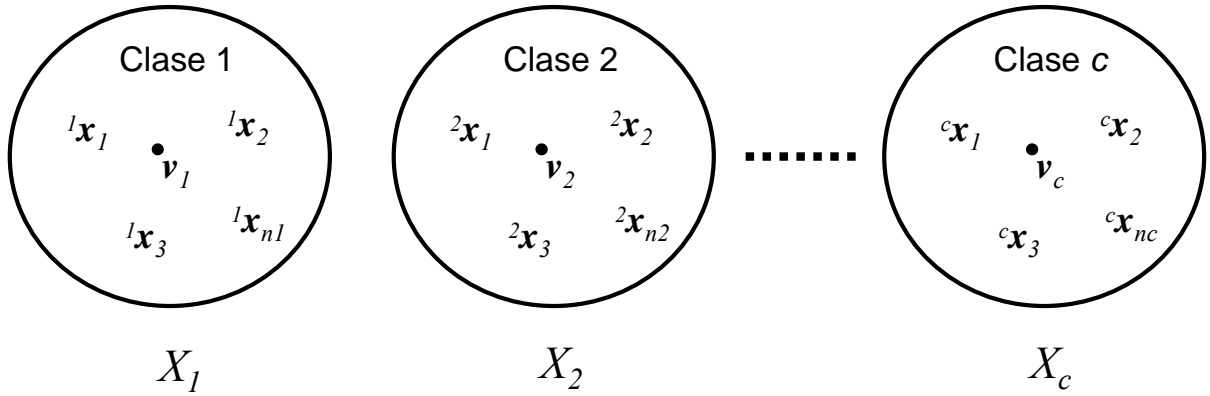


Figura 3.4 Distribución de las muestras y centros por clases tras la validación de la partición.

Centros de los clústeres tras la partición fuzzy

Según la distribución de la figura 3.4, las  $N$  muestras del conjunto  $X$  se distribuyen en  $c$  clases, donde las clases contienen  $n_1, n_2, \dots, n_c$  muestras cada una de ellas. Cada clase tiene asignado un vector  $v_1, v_2, \dots, v_c$  que representa el centro de la clase a la que pertenece, obtenido como ya se ha mencionado mediante la ecuación (2.5). El resultado de la partición es:  $N = n_1 + n_2 + \dots + n_c$  teniendo  $X = X_1 \cup X_2 \cup \dots \cup X_c$ .

#### **b) Estima de Bayes**

Conocida la partición, se procede a obtener mediante Bayes los parámetros representativos de cada una de las clases, a saber las medias  $m_j$  y las matrices de covarianza  $C_j$ . Estos parámetros se obtienen mediante la ecuación (2.12).

En la figura 3.5 se muestran los parámetros obtenidos

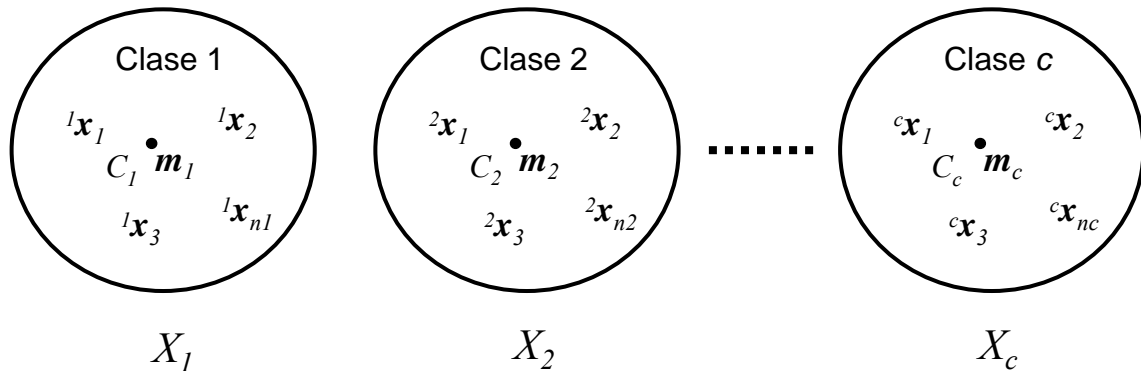


Figura 3.5 Centros de los clústeres y matrices de covarianza de cada clúster tras la estima de Bayes

### c) Estima de Parzen

En realidad, la estima de Parzen sólo necesitará tener en cuenta las matrices de covarianza  $C_j$  junto con las muestras asignadas a cada clase.

### d) Almacenamiento en la Base de Conocimiento (BC)

En la Base de Conocimiento (BC) se almacenan los siguientes datos:

- Los centros  $v_j$  procedentes del clasificador fuzzy
- Los centros  $m_j$  y las matrices de covarianza  $C_j$
- Las muestras según su distribución por clases  $^c x_i$

## 3.3.2. Fase de clasificación

El problema que se plantea en esta fase de clasificación consiste en determinar a qué clase pertenece una muestra dada  $x_s$  que se presenta como consecuencia del procesamiento de una nueva imagen, sobre la cual se pretende establecer el tipo de texturas que se han de detectar.

El procedimiento consiste en que cada una de los métodos empleados recupera exactamente lo que necesita de la BC.

### a) Clasificación mediante Bayes



Siguiendo el esquema de la figura 3.1, el clasificador de Bayes extrae de la BC tanto los centros de cada una de las clases como las matrices de covarianza, es decir:  $\mathbf{m}_j$  y  $C_j$ .

Con esta información recuperada y mediante la ecuación (2.13), que reproducimos aquí por simplicidad, ecuación (3.4), se obtiene la probabilidad de pertenencia a cada una de las clases. Concretamente, mediante la ecuación (3.4) indicamos que la probabilidad de Bayes ( $p_b$ ) de que la muestra  $\mathbf{x}_s$  pertenezca a la clase  $w_j$  representada por sus parámetros  $w_j \equiv (\mathbf{m}_j, C_j)$  es la que se proporciona a continuación

$$p_b(\mathbf{x}_s | \mathbf{m}_j, C_j) = \frac{1}{(2\pi)^{d/2} |C_j|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x}_s - \mathbf{m}_j)^T C_j^{-1} (\mathbf{x}_s - \mathbf{m}_j) \right\} \quad (3.4)$$

De esta forma calcularemos tantas  $p_b$  como clases tengamos, es decir  $j = 1, 2, \dots, c$ . Dado que los valores  $p_b$  pueden exceder el límite de +1 y por tanto sobrepasar el valor límite superior de los valores de probabilidad, es práctica común derivada de la propia teoría de Bayes proceder a realizar una normalización, que en realidad se trata de la aplicación del cálculo de la función de densidad de probabilidad mixta. Por consiguiente, aplicando dicha normalización obtenemos finalmente,

$$p_b(\mathbf{x}_s | \mathbf{m}_j, C_j) = \frac{p_b(\mathbf{x}_s | \mathbf{m}_j, C_j)}{\sum_{i=1}^c p_b(\mathbf{x}_s | \mathbf{m}_i, C_i)} \quad (3.5)$$

De esta forma se consigue restringir los valores de probabilidad al rango  $[0, +1]$ , que será un requisito imprescindible para llevar a cabo posteriormente la operación de agregación mediante los operadores del mismo nombre de naturaleza borrosa.

En resumen, si se dispone de las clases  $w_1, w_2, \dots, w_c$  como posibles para clasificar la muestra dada, se dispondrá igualmente del mismo número de probabilidades de pertenencia a cada una de las clases según la ecuación (3.5), obteniendo finalmente la serie de valores dados en la ecuación (3.6).

$$\begin{aligned}
p_b(\mathbf{x}_s | w_1) &\equiv p_b(\mathbf{x}_s | \mathbf{m}_1, C_1); \\
p_b(\mathbf{x}_s | w_2) &\equiv p_b(\mathbf{x}_s | \mathbf{m}_2, C_2), \dots, p_b(\mathbf{x}_s | w_c) \equiv p_b(\mathbf{x}_s | \mathbf{m}_c, C_c)
\end{aligned} \tag{3.6}$$

Cuando se utiliza el clasificador de Bayes de forma individual, es decir sin ser combinado con ningún otro, el criterio para determinar a qué clase pertenece  $\mathbf{x}_s$  consiste en seleccionar la máxima probabilidad de entre todas las obtenidas y asignar  $\mathbf{x}_s$  a la clase asociada con este valor máximo de probabilidad. Formalmente esto se expresa como sigue,

$$\mathbf{x}_s \in w_j \mid p_b(\mathbf{x}_s | w_j) > p_b(\mathbf{x}_s | w_k) \quad \forall k \neq j \quad j = 1, 2, \dots, c \tag{3.7}$$

Este criterio será el utilizado para la comparación del método propuesto en este trabajo frente al clasificador de Bayes utilizado de forma individual.

#### ***b) Clasificación mediante la técnica Fuzzy***

El clasificador fuzzy recibe los centros de los clústeres  $\mathbf{v}_j$  almacenados en la BC. Dada la muestra de entrada  $\mathbf{x}_s$  y utilizando la ecuación (2.6) determinamos el grado de pertenencia  $\mu_{sj}$ , de dicha muestra a cada una de las clases  $j$ . Esta operación se lleva a cabo mediante la ecuación citada que reproducimos a continuación por simplicidad,

$$\mu_{sj} = \frac{\left( \frac{1}{\|\mathbf{x}_s - \mathbf{v}_j\|_G^2} \right)^{2/m-1}}{\sum_{h=1}^c \left( \frac{1}{\|\mathbf{x}_s - \mathbf{v}_h\|_G^2} \right)^{2/m-1}} \quad j = 1, \dots, c; k = 1, \dots, n \tag{3.8}$$

Realizado este proceso, se determinan los grados de pertenencia de la citada muestra a cada una de las clases. Resultando, de la propia definición de los grados de pertenencia que los mismos están restringidos al rango de valores  $[0, +1]$  como en el caso de la probabilidad de Bayes.

Del mismo modo, como en el caso del clasificador de Bayes, si se dispone de las clases  $w_1, w_2, \dots, w_c$  como posibles para clasificar la muestra dada, se dispondrá igualmente del mismo número de grados de pertenencia a cada una de las clases según la ecuación (3.8), obteniendo finalmente la secuencia de valores dados en (3.9).

$$\mu_{s1} \equiv \mu(\mathbf{x}_s, w_1), \mu_{s2} \equiv \mu(\mathbf{x}_s, w_2), \dots, \mu_{sc} \equiv \mu(\mathbf{x}_s, w_c) \quad (3.9)$$

Cuando se utiliza el clasificador Fuzzy de forma individual, es decir sin ser combinado con ningún otro, el criterio para determinar a qué clase pertenece  $\mathbf{x}_s$  consiste en seleccionar el máximo valor de los grados de pertenencia de entre todos los obtenidos y asignar  $\mathbf{x}_s$  a la clase asociada con este valor máximo. Formalmente esto se expresa como sigue,

$$\mathbf{x}_s \in w_j \mid \mu_{sj} > \mu_{sk} \quad \forall k \neq j \quad j = 1, 2, \dots, c \quad (3.10)$$

Este criterio será el utilizado para la comparación del método propuesto en este trabajo frente al clasificador Fuzzy utilizado de forma individual.

### ***c) Clasificación mediante Parzen***

El clasificador de Parzen recibe las matrices de covarianza  $C_j$  procedentes de la BC además de todas las muestras disponibles, también almacenadas en la BC y distribuidas por clases. Con estos datos y teniendo en cuenta la ecuación (2.17) se obtiene una probabilidad de pertenencia de la muestra a clasificar  $\mathbf{x}_s$  a cada una de las diferentes clases disponibles. En este caso hay que tener en cuenta que los números de muestras asignados a cada clase serán  $n_1$  para la clase  $w_1$ ,  $n_2$  par la clase  $w_2$  y así sucesivamente  $n_c$  para la clase  $w_c$ . Por tanto la probabilidad de pertenencia de  $\mathbf{x}_s$  a la clase  $w_j$  se obtiene mediante la citada ecuación (2.17), que reproducimos a continuación en (3.11) de forma particularizada para el caso que nos ocupa.

$$p_p(\mathbf{x}_s \mid w_j) = \frac{1}{n_j} \sum_{i=1}^{n_j} \left\{ \frac{1}{(2\pi)^{p/2} |C_j|^{1/2} h^p} \exp \left[ -\frac{1}{2h^2} (\mathbf{x}_s - \mathbf{x}_i)^t C_j^{-1} (\mathbf{x}_s - \mathbf{x}_i) \right] \right\} \quad (3.11)$$

Al igual que en el caso del clasificador de Bayes, de esta forma calcularemos tantas  $p_p$  como clases tengamos, es decir  $j = 1, 2, \dots, c$ . Dado que los valores  $p_p$  pueden exceder el límite de +1 y por tanto sobrepasar el límite superior de los valores de probabilidad, es práctica común derivada de la propia teoría de Bayes proceder a realizar una normalización, que en realidad se trata de la aplicación del cálculo de la función de densidad de

probabilidad mixta. Por consiguiente, aplicando dicha normalización obtenemos finalmente,

$$p_p(\mathbf{x}_s | w_j) = \frac{p_p(\mathbf{x}_s | w_j)}{\sum_{i=1}^c p_b(\mathbf{x}_s | w_i)} \quad (3.12)$$

De esta forma se consigue restringir los valores de probabilidad al rango  $[0, +1]$ , que será un requisito imprescindible para llevar a cabo posteriormente la operación de agregación mediante los operadores del mismo nombre de naturaleza borrosa.

En resumen, como en los dos casos anteriores, si se dispone de las clases  $w_1, w_2, \dots, w_c$  como posibles para clasificar la muestra dada, se dispondrá igualmente del mismo número de probabilidades de pertenencia a cada una de las clases según la ecuación (3.12), es decir:

$$p_p(\mathbf{x}_s | w_1), p_p(\mathbf{x}_s | w_2), \dots, p_p(\mathbf{x}_s | w_c) \quad (3.13)$$

Como en los casos anteriores, cuando se utiliza el clasificador de Parzen de forma individual, es decir sin ser combinado con ningún otro, el criterio para determinar a qué clase pertenece  $\mathbf{x}_s$  consiste en seleccionar la máxima probabilidad de entre todas las obtenidas y asignar  $\mathbf{x}_s$  a la clase asociada con este valor máximo de probabilidad. Formalmente esto se expresa como sigue,

$$\mathbf{x}_s \in w_j \mid p_p(\mathbf{x}_s | w_j) > p_p(\mathbf{x}_s | w_k) \quad \forall k \neq j \quad j = 1, 2, \dots, c \quad (3.14)$$

Este criterio será el utilizado para la comparación del método propuesto en este trabajo frente al clasificador de Parzen utilizado de forma individual.

### 3.4. Hibridación

Como colofón a la fase de clasificación y tras procesar la muestra  $\mathbf{x}_s$  por medio de los tres clasificadores mencionados (Bayes, Fuzzy y Parzen), se obtienen o bien una serie de probabilidades de pertenencia a las diferentes clases para la muestra dada (Bayes y Parzen) o bien los correspondientes grados de pertenencia. Es decir, las salidas proporcionadas por los diferentes clasificadores se pueden sintetizar según el esquema proporcionado por la tabla 3.1 dada a continuación.

<b>Clasificador</b>	<b>Clase 1 (<math>w_1</math>)</b>	<b>Clase 2 (<math>w_2</math>)</b>		<b>Clase <math>c</math> (<math>w_c</math>)</b>
<b>Bayes</b> ecuación (3.6)	$p_b(\mathbf{x}_s   w_1)$	$p_b(\mathbf{x}_s   w_2)$	...	$p_b(\mathbf{x}_s   w_c)$
<b>Fuzzy</b> ecuación (3.9)	$\mu(\mathbf{x}_s, w_1)$	$\mu(\mathbf{x}_s, w_2)$	...	$\mu(\mathbf{x}_s, w_c)$
<b>Parzen</b> ecuación (3.13)	$p_p(\mathbf{x}_s   w_1)$	$p_p(\mathbf{x}_s   w_2)$	...	$p_p(\mathbf{x}_s   w_c)$

Tabla 3.1 Salidas proporcionadas por los clasificadores Bayes, Fuzzy y Parzen tras procesar la muestra  $\mathbf{x}_s$

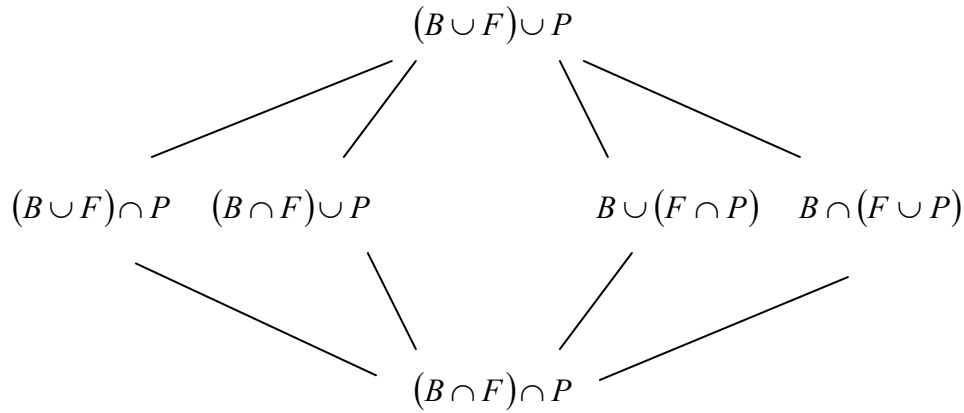
Teniendo en cuenta la discusión realizada en la sección 2.4.2 el problema que se plantea en este momento es determinar finalmente mediante la operación de agregación fuzzy cuál será la decisión final tomada respecto de la asignación de  $\mathbf{x}_s$  a una de las clases.

Teniendo en cuenta las propiedades de las diferentes lógicas (Zadeh y lógica del producto), se elige la del **producto** por sus más convenientes propiedades. Las combinaciones o agregaciones propuestas se sintetizan en la siguiente tabla 3.2. La simbología es la siguiente  $B \equiv$  Bayes,  $F \equiv$  Fuzzy y  $P \equiv$  Parzen, teniendo en cuenta que para la agregación Bayes utilizará  $p_b(\mathbf{x}_s | w_j)$ , Fuzzy  $\mu(\mathbf{x}_s, w_j)$  y Parzen  $p_p(\mathbf{x}_s | w_j)$ .

Tabla 3.2 Hibridaciones

<b>Hibridaciones</b>	<b>Lógica del producto</b>
<b>H1</b>	$(B \cup F) \cup P$
<b>H2</b>	$(B \cup F) \cap P$
<b>H3</b>	$(B \cap F) \cup P$
<b>H4</b>	$(B \cap F) \cap P$
<b>H5</b>	$B \cup (F \cap P)$
<b>H6</b>	$B \cap (F \cup P)$

Un esquema representativo del contenido de la tabla 3.2 se muestra a continuación.



Se han realizado diferentes experimentos con los operadores de agregación fuzzy descritos en la tabla 2.2 sin llegar a conseguir mejoras sustanciales respecto de las combinaciones dadas en la tabla 3.2

El proceso de decisión final consiste en realizar la agregación de los valores entre paréntesis, para posteriormente proceder a la agregación del resultado así obtenido con el valor que queda fuera del paréntesis. Para la agregación H1 el proceso se sintetiza como sigue:

1. Dada la clase  $w_j$  se realizan las siguientes operaciones: agregar Bayes y Fuzzy:

$$^1A_j = p_b(\mathbf{x}_s | w_j) \cup \mu(\mathbf{x}_s, w_j) \quad (3.15)$$

agregar el resultado previo con Parzen:

$$^2A_j = ^1A_j \cup p_p(\mathbf{x}_s | w_j) \quad (3.16)$$

2. La decisión final se toma eligiendo el máximo de los diferentes  $^2A_j$  obtenidos, es decir:

$$^2A_J = \max_{j=1}^c \{ ^2A_j \} \quad (3.17)$$

Por tanto, según lo anterior  $\mathbf{x}_s$  pertenece a la clase que indique el índice  $J$ , según la ecuación (3.17).

La justificación para elegir el criterio de máximo en la ecuación (3.17) y no otro se fundamenta en el siguiente razonamiento. Sabemos que tanto el clasificador Bayes como Fuzzy proporcionan valores de probabilidad de pertenencia de la muestra  $x_s$  a cada una de las clases, como la probabilidad varía en el rango  $[0,1]$  se establece que cuanto mayor sea esa probabilidad mayor es el grado de pertenencia a la clase dada. Es decir un valor de la unidad significa máximo valor de pertenencia, mientras que un valor nulo indica justamente lo contrario.

El razonamiento anterior es igualmente válido para el clasificador Fuzzy, ya que en este caso son valores de pertenencia a las clases lo que se obtiene variando igualmente en el rango  $[0,1]$  con idénticos criterios en cuanto a la pertenencia a las clases se refiere, esto es, máxima pertenencia para valores de la unidad y mínima para valores nulos.

Las seis combinaciones descritas en la tabla 6.2 se basan en operaciones algebraicas entre conjuntos, generalizadas al caso borroso. Para explicarlo, definimos los siguientes conjuntos borrosos:

Sea  $F_c$  un conjunto borroso sobre el conjunto de píxeles que obtiene para cada píxel un grado de pertenencia dado por el valor asignado por el método Fuzzy para dicho píxel en el clúster  $c$ .

Sea  $B_c$  un conjunto borroso sobre el conjunto de píxeles que obtiene para cada píxel un grado de pertenencia dado por el valor asignado por el método de Bayes para dicho píxel en el clúster  $c$ .

Sea  $P_c$  un conjunto borroso sobre el conjunto de píxeles que obtiene para cada píxel un grado de pertenencia dado por el valor asignado por el método de Parzen para dicho píxel en el clúster  $c$ .

Estos tres conjuntos borrosos se definen sobre un mismo universo de píxeles, así pues sabemos realizar operaciones algebraicas de conjuntos borrosos utilizando operadores de t-normas y t-conormas.

Entonces el método H1 se puede basar en la unión de dichos tres conjuntos, y el resultado dado en (3.16) se puede entender que  ${}^2A_j$  es un conjunto borroso definido como según la

siguiente combinación:  $F_c \cup B_c \cup P_c$ . En dicho método, para que un píxel pertenezca a un clúster del método híbrido basta con que dicho clúster pertenezca a alguno de los métodos, entendiendo ahora que dicha pertenencia es borrosa y se da mediante un grado de pertenencia. Dicha unión se puede hacer de muchas maneras, eligiendo diferentes operadores de t-conorma con diferentes propiedades, pero en el caso de que sea nítida, todas son iguales y generalizan la unión algebraica de conjuntos.

El método H6 se puede basar en la intersección de dichos tres conjuntos, y se puede entender que  $^2A_j$  es un conjunto borroso definido como  $F_c \cap B_c \cap P_c$ . En dicho método, para que un píxel pertenezca a un clúster del método tiene que pertenecer a dicho clúster en todos los métodos, entendiendo ahora que dicha pertenencia es borrosa y se da mediante un grado de pertenencia. Dicha intersección se puede hacer de muchas maneras, eligiendo diferentes operadores de t-norma con diferentes propiedades, pero en el caso de que sea nítida, todas son iguales y generalizan la intersección algebraica de conjuntos.

Nótese que el orden de operar la unión o intersección no es relevante, debido a la propiedad asociativa de los operadores de t-norma y t-conorma.

Ahora, la unión aumenta los grados de pertenencia de los conjuntos borrosos definidos para los tres métodos, mientras que la intersección los disminuye. Por ese motivo hemos estudiado otras 4 formas de operar algebraicamente los conjuntos definidos para los 3 métodos que no sean comparables a ninguno de ellos (algunos grados mayores y otros menores), por ejemplo, usando un operador de unión de dos e intersección con el tercero.

La lógica de Zadeh (como la usada para el método del máximo y mínimo) no nos ha parecido adecuada, pues los valores de pertenencia de los conjuntos borrosos (uno para cada clúster) definido en el método híbrido, al final resulta ser exactamente el de uno de los métodos, por lo que más que una hibridación es una selección de los grados de pertenencia, que indica la pertenencia de cada píxel a un clúster, para uno de los tres métodos, aunque puede ser diferente el método elegido para cada píxel y clúster.

La lógica de Lukasiewicz en el caso de la unión sólo funciona para agregar valores pequeños (para valores grandes toma rápidamente el valor de 1) y en el caso de la



intersección sólo funciona para agregar grados de pertenencia grandes (para valores pequeños toma rápidamente el valor de 0).

La lógica del producto no presenta los inconvenientes de las citadas lógicas, y consigue valores que realmente son híbridos, funcionando adecuadamente en la mayoría de los casos.



## 4. Análisis de resultados

### 4.1. Objetivos del análisis

En este punto se trata de verificar y validar la estrategia propuesta. Al tratarse de un clasificador híbrido que realiza un proceso de aprendizaje previo a la clasificación, se plantean los siguientes tres objetivos de verificación:

- Comportamiento del método híbrido frente a clasificadores simples (no híbridos)
- Comportamiento del método frente a otras estrategias de hibridación
- Comportamiento del método a medida que se incrementa el aprendizaje

Para el primer objetivo se eligen como estrategias simples los métodos individuales que se utilizan en el clasificador híbrido propuesto, a saber: Bayes, Fuzzy y Parzen.

Para verificar el segundo objetivo se utilizan los tres métodos de hibridación de clasificadores descritos en la sección 2.3, a saber: Reglas de máximo, Reglas de mínimo y Regla de media.

Finalmente para la consecución del tercer objetivo se diseñan varias etapas de aprendizaje de forma que en cada una de ellas se verifique el comportamiento del método frente a las etapas anteriores.

### 4.2. Descripción de las imágenes utilizadas

Se dispone de un conjunto de 20 imágenes aéreas adquiridas durante el mes de Mayo de 2006 en la región de Abadía en la provincia de Lugo (España). Se trata de imágenes digitales multiespectrales con una dimensión de 512x512 píxeles. Por multiespectral se entiende imágenes de color con los tres canales correspondientes al espectro visible, a saber: Rojo (R), Verde (G) y Azul (B). Las imágenes se capturaron en diferentes vuelos y

días del mencionado mes. Las imágenes con las que se han llevado a cabo los experimentos son cortesía del Servicio Territorial de Galicia cedidas por la empresa Dimap, como ya se ha indicado anteriormente.

De las 20 imágenes disponibles se eligen 4 aleatoriamente con el fin de seleccionar sobre las mismas el conjunto inicial de muestras de entrenamiento con el que se inicia el proceso. Este conjunto se denomina S1. Dado que la estrategia propuesta es *no supervisada*, estas muestras no están inicialmente clasificadas. La aplicación implementada permite seleccionar las muestras bien por medio del ratón o de forma automática leyendo las imágenes de entrada directamente.

Otro conjunto formado por otras 4 imágenes se seleccionan igualmente de forma aleatoria con el fin de verificar la mejora del proceso de aprendizaje. Este conjunto de muestras se denomina S0 y su elección se lleva a cabo igualmente de forma aleatoria.

Las 12 imágenes restantes se dividen en tres grupos o conjuntos (S1 S2 y S3) cada uno formado por 4 imágenes. La distribución de las imágenes en estos grupos se hace también de forma aleatoria con el fin de no condicionar ningún posible resultado futuro. Las pruebas se llevan a cabo mediante el proceso descrito a continuación.

En la figura 4.1 se muestran dos imágenes representativas de los conjuntos mencionados. En (a) una imagen del primer grupo utilizado como grupo inicial. En (b) una imagen perteneciente al grupo S0.

En la figura 4.2 se muestran tres imágenes representativas de los conjuntos de prueba S1, S2 y S3.

Como puede apreciarse, sobre las mismas aparecen básicamente varios tipos de texturas entre las que destacan:

- masas de arbustos en oscuro
- praderas y campos verdes
- campos de cultivo en barbecho
- edificios y construcciones
- caminos de tierra



(a)

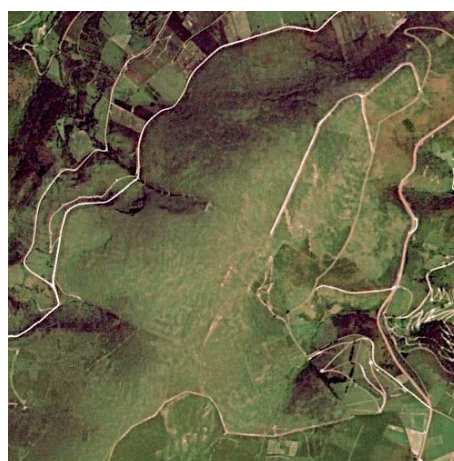


(b)

Figura 4.1 Imágenes representativas de los conjuntos utilizados: (a) imagen de entrenamiento inicial; (b) imagen representativa del conjunto S0



(a)



(b)



(c)

Figura 4.2 Imágenes representativas de los conjuntos utilizados: (a) S1; (b) S2 y (c) S3

### 4.3. Diseño de una estrategia de prueba

La implementación se ha realizado en Matlab, que es un producto de la compañía The Mathworks <http://www.mathworks.com/products/matlab/>. Se trata de un lenguaje científico interpretado. Debido a este hecho el coste computacional de los procesos implementados en Matlab es superior a otros lenguajes de distinta naturaleza. No obstante, el objetivo de este trabajo no consiste en estudiar los tiempos de cómputo sino el comportamiento del método propuesto. En el Anexo I se muestran las ventanas de diálogo más relevantes para el proceso de aprendizaje y clasificación.

Las pruebas se llevan a cabo siguiendo los pasos que se explican a continuación, sabiendo que disponemos de los siguientes conjuntos: SI (inicial), S0, S1, S2 y S3.

#### **PASO 0** (inicial):

Para cada imagen del conjunto de 4 imágenes SI seleccionadas como iniciales se realiza un submuestreo por 4. Este proceso se lleva a cabo tomando de la imagen original una fila de cada cuatro, así como también una columna de cada cuatro. El resultado es que de cada imagen de  $512 \times 512$  se toman  $128 \times 128$  muestras, obteniendo en total  $4 \times 128 \times 128 = 65536$  muestras.

Con todas estas muestras, se inicia el proceso de partición de las mismas en  $c$  clases, para ello se realiza un proceso iterativo comenzando con  $c = 2$ , fijando un límite máximo cuyo valor es  $c = 8$ . El objetivo consiste en establecer una partición válida. El procedimiento se sintetiza como sigue, según se describe en la sección 3.3.1a,

1. Desde  $c = 2$  hasta  $c = 8$  realizar el proceso descrito a continuación
2. Obtener el coeficiente de partición  $CP$  para cada partición obtenida.
3. Considerar como particiones válidas solamente las que cumplan que  $CP > 0.80$ . Si no hubiese ninguna partición válida se probarían con más valores de  $c$ . Este caso no se ha dado nunca en nuestros experimentos.
4. De entre todas las particiones válidas seleccionar la que posea un mayor valor del  $CP$ , determinando para ella el valor de  $c$ .
5. Conocido el número de clústeres  $c$ , se procede a la distribución de las muestras en los correspondientes clústeres y se determina mediante el procedimiento también descrito en la sección 3.3.1 del apartado a) los centros de cada clúster.

6. Las muestras distribuidas por clases son proporcionadas al estimador Bayesiano para realizar el proceso descrito en la sección 3.3.1 del apartado *b*). De esta manera se obtienen los centros de las clases  $m_j$  junto con sus matrices de covarianza  $C_j$ .
7. Las mismas muestras distribuidas en las mismas clases son proporcionadas al clasificador de Parzen para proceder a la estima de la función de densidad de probabilidad según el procedimiento descrito en la sección 3.3.1 del apartado *c*).

#### **PASO 1:**

Dadas las imágenes de los conjuntos S0 y S1 se procede a la clasificación de cada píxel en cada imagen mediante los clasificadores Bayes, Fuzzy y Parzen tal y como se describe en la sección 3.3.2 según los apartados *a*), *b*) y *c*).

Con las muestras clasificadas procedentes de ambos conjuntos se procede al cómputo de los errores obtenidos.

Con las muestras clasificadas procedentes del conjunto S1 se realiza un nuevo proceso de entrenamiento (aprendizaje), sin variar el número de clases. Es decir se estiman de nuevo los centros de las clases según el procedimiento Fuzzy, los centros y las matrices de covarianza según el procedimiento de Bayes y se estima la función de densidad de probabilidad según Parzen.

Las muestras clasificadas procedentes de S0 no se utilizan para las nuevas estimas, esto es así con el fin de comprobar cómo evoluciona el error de clasificación con estas muestras a medida que el aprendizaje aumenta.

#### **PASO 2:**

En este paso se procede del mismo modo que en el Paso 1, si bien utilizando los conjuntos S0 y S2. Ahora los resultados de S2 se utilizan para realizar un nuevo proceso de entrenamiento.

#### **PASO 3:**

En este paso se procede del mismo modo que en los Pasos 1 y 2, si bien utilizando los conjuntos S0 y S3. Ya no es necesario realizar más procesos de entrenamiento.

Como puede fácilmente deducirse de lo expuesto anteriormente, en los pasos 1 y 2 se añaden nuevas muestras para realizar un nuevo proceso de entrenamiento. El número de muestras añadidas en cada uno de estos dos pasos resulta ser  $4 \times 512 \times 512 = 10486576$ , que se corresponde con el resultado de multiplicar el número de píxeles de cada imagen por el número de imágenes. El número total de muestras de entrenamiento suministradas durante el proceso de prueba resulta ser por tanto  $2 \times 4 \times 512 \times 512 = 2097152$ .

El proceso se sintetiza según la figura 4.3, en ella se muestra el proceso en las dos fases contempladas: clasificación y entrenamiento. Inicialmente se realiza un primer entrenamiento para obtener la partición válida y los parámetros derivados del aprendizaje:  $v_j$ ,  $m_j$  y  $C_j$ . Posteriormente, en los pasos 1 y 2 se realiza un proceso de clasificación seguido de un proceso de entrenamiento, que permite actualizar los parámetros  $v_j$ ,  $m_j$  y  $C_j$  al incorporar las nuevas muestras procedentes de las clasificaciones de S1 y S2. El orden de ejecución sigue la secuencia de pasos mostrada por los números marcados.

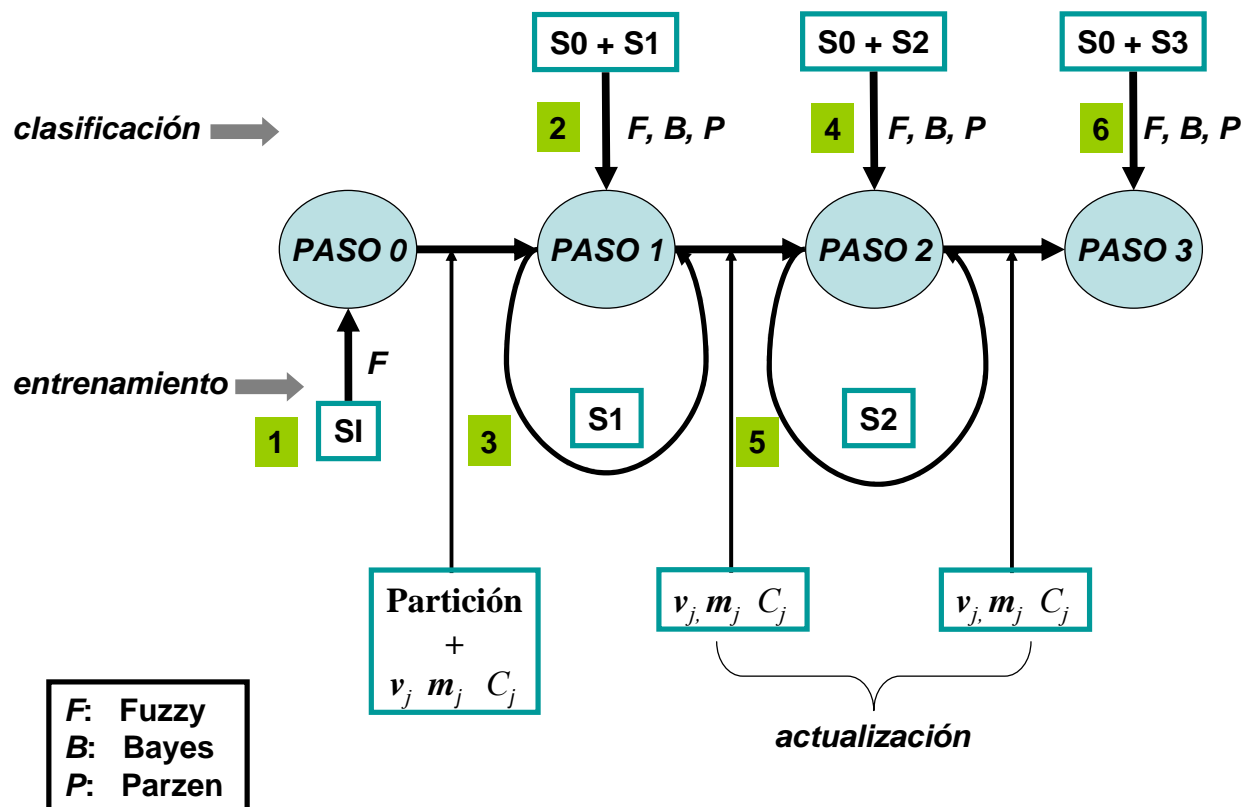


Figura 4.3 Síntesis de la estrategia de prueba



## 4.4. Análisis de resultados

### 4.4.1. Determinación del número de clases

El primer objetivo que se plantea consiste en determinar el número de clases que va a poseer la partición inicial. Es decir se trata de validar la partición inicial.

La figura 4.4 muestra el comportamiento del coeficiente de partición frente al número de clústeres o clases. Puede observarse que existen dos valores del coeficiente de partición que superan el valor de umbral prefijado en 0.8. Estos se corresponden con los valores de  $c$  para 4 y 5, con valores respectivos de 0.86 y 0.82. Como quiera que en este caso el máximo valor se obtiene para  $c = 4$ , este valor se elige finalmente como el número de clústeres para el proceso.

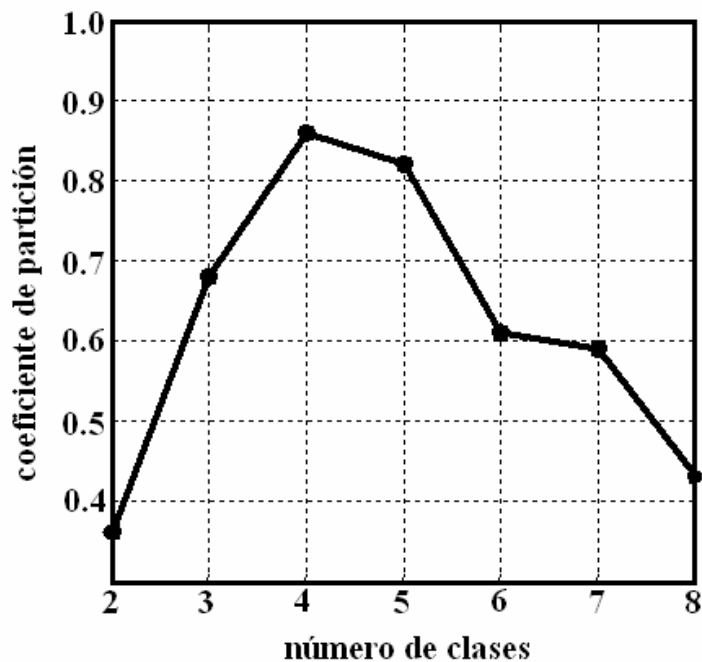


Figura 4.4 Evolución del coeficiente de partición frente al número de clases

Puede observarse de la figura 4.3 que para valores de números de clases distintos a cuatro y cinco no se consigue un valor aceptable según el criterio establecido de umbral 0.8.

Cuando la partición tiene un número de clases bajo significa que existen muchas muestras que son asignadas a clases de una forma muy forzada, lo que se traduce en que las mismas se sitúan lejos de los centros de los clústeres y por tanto generan valores de los grados de pertenencia bajos.

En el caso de un número de clases alto, existen muestras que se asignan a un clúster pero que en realidad podrían estar asignadas a cualquier otro clúster próximo. Esto se traduce en el hecho de que existirán muestras con elevados grados de pertenencia para clases diferentes.

#### 4.4.2. Criterio de validación

Como se ha indicado durante la introducción no existe una base de datos con la que poder verificar los resultados de la clasificación para el tipo de imágenes propuestas. Se hace necesario establecer para cada una de las imágenes clasificadas una base de comparación. Esto es lo que técnicamente se denomina “ground-truth”. El objetivo por tanto, se centra ahora en establecer un procedimiento para poder llevar a cabo el proceso de validación. A continuación ilustramos este proceso sirviéndonos como ejemplo ilustrativo la imagen dada en la figura 4.1(b) perteneciente al conjunto S0 descrito con anterioridad.

##### *a) Extracción del número de clústeres inicial*

Fijado el número de clústeres por el procedimiento descrito en la sección 4.4.1, se utiliza el clasificador de Bayes para establecer una partición inicial. En la figura 4.5(a) se reproduce la imagen de la figura 4.1(b) para mayor facilidad.

En la imagen 4.5(b) se muestra el resultado de la clasificación según Bayes. En la imagen de la figura 4.6 se muestra la correspondencia existente entre las clases de la imagen real y la imagen etiquetada y coloreada.

Puede observarse cómo la correspondencia resulta ser la siguiente:

- **Amarillo** con masas de vegetación boscosa
- **Azul** con suelos no cultivados
- **Verde** con vegetación de pradera y cultivos
- **Rojo** con construcciones e infraestructuras humanas

A la vista de la imagen etiquetada o coloreada puede observarse realmente que las diferentes texturas aparecen entremezcladas, no resultando fácil llevar a cabo una distinción o separación nítida en lo que a las mencionadas texturas se refiere. Por este

motivo no resulta fácil establecer el conjunto de imágenes de validación por clases con el fin de determinar el conjunto base o ground-truth anteriormente mencionado.

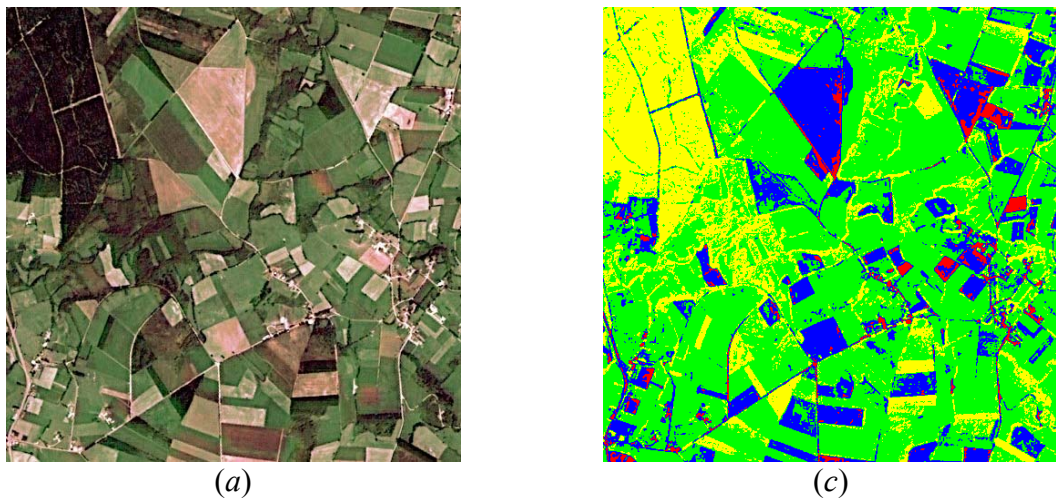


Figura 4.5 (a) imagen original representativa del grupo S0; (b) clases etiquetadas mediante Bayes



Figura 4.6 Correspondencia entre las clases reales y las etiquetas (colores) para las clases

#### ***b) Construcción del conjunto base de pruebas o “ground-truth”***

Este proceso consiste en tomar como referencia las imágenes separadas de cada clase, las cuales se obtienen aislando cada una de las clases obtenidas mediante el clasificador de Bayes. Estas imágenes se almacenan en forma de imágenes binarias donde para cada una de las clases, los píxeles pertenecientes a la misma se almacenan en blanco, es decir con valores de intensidad de 255 y el resto en negro, en este caso con valores de intensidad 0.

En la figura 4.7 se muestra la serie correspondientes a los cuatro clústeres definidos en la figura 4.6 con el orden de clases mostrado en la imagen de esta figura. Es decir, los clústeres 1 a 4 se corresponden con las imágenes (a)-(d). Estas imágenes así obtenidas se

retocan manualmente teniendo en cuenta cada imagen original, de esta forma se obtiene la serie (e)-(h) que constituye ya el ground-truth deseado.

Este proceso se realiza para todos los conjuntos de imágenes procesadas.

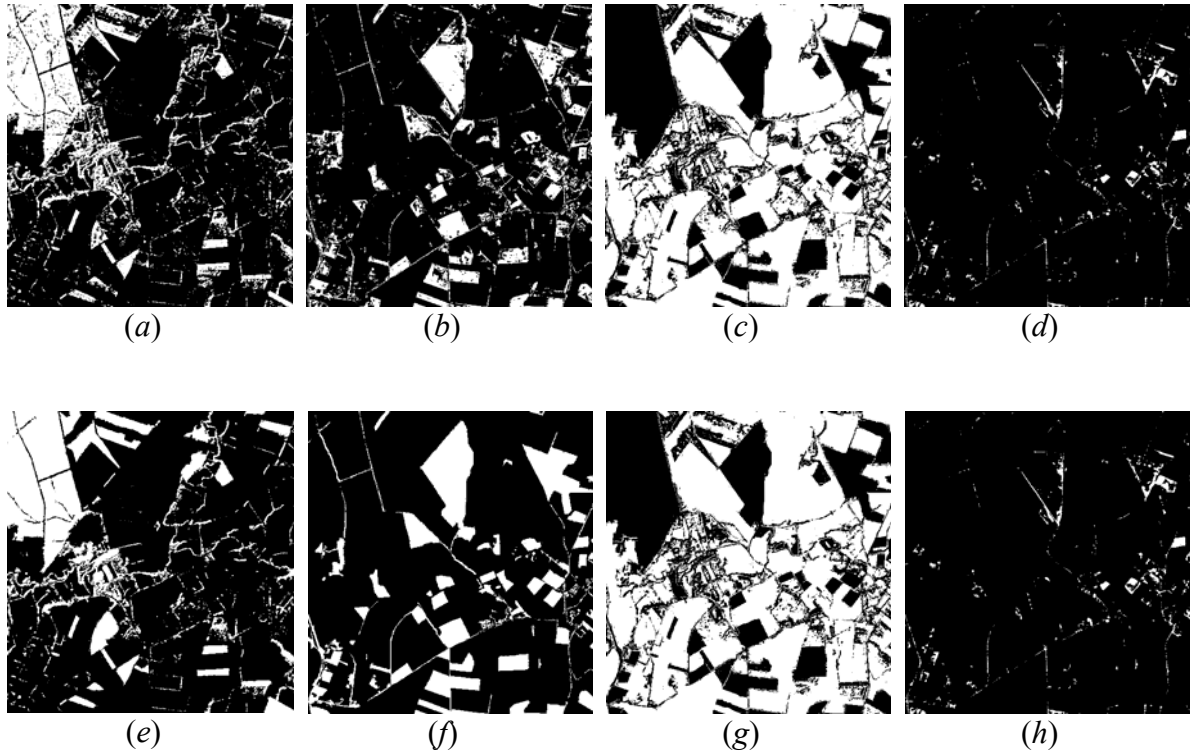


Figura 4.7 Imágenes de clases separadas mostrando desde la clase 1 a la 4 en orden comenzando por la izquierda: (a) a (d) serie original; (e) a (h) serie manualmente retocada

En relación con lo expuesto anteriormente, y desde el punto de vista de la validación, el criterio finalmente establecido consiste en suponer que un clasificador obtiene un error de clasificación nulo cuando el resultado que genera es exactamente el correspondiente a las imágenes retocadas. En el caso que nos ocupa, las correspondientes al conjunto mostrado en la figura 4.7 (e)-(h).

En el Anexo II se muestran varios ejemplos de imágenes clasificadas mediante el procedimiento híbrido descrito en este trabajo.

#### 4.4.3. Resultados obtenidos

Seguidamente se lleva a cabo el proceso de verificación y validación según las cuatro etapas que se describen a continuación:

1. Dado que el método híbrido propuesto contempla la posibilidad de realizar diferentes hibridaciones, tal y como se describe en la tabla 3.2, la primera fase consiste en determinar cuál de las propuestas de hibridación (H1, H2, H3, H4, H5, H6) resulta más efectiva. Esta prueba, por tanto identifica el mejor método de combinación o hibridación.
2. Fijado el método híbrido más efectivo se compara éste frente a los clasificadores individuales, a saber Bayes (BY), Fuzzy (FU) y Parzen (PA).
3. Dado que el método propuesto en este trabajo es un método híbrido, el siguiente paso consiste en verificar su comportamiento frente a tres de los métodos más clásicos utilizados en la literatura y descritos en la sección 2.3 y la tabla 2.1, a saber: Regla del máximo (MA), Regla del Mínimo (MI), Regla de la media (ME) y Votador mayoritario (VM).
4. Finalmente, de acuerdo con el método explicado en la sección 3.1 relativo a los antecedentes de este trabajo, se presentan los resultados relativos a la combinación del clasificador Fuzzy y el Bayesiano (FB) y por consiguiente sin la intervención de Parzen.

En todos los casos la validación se lleva a cabo en los tres pasos mencionados en la sección 4.3 con el fin de verificar la influencia del número de muestras empleadas en el proceso de entrenamiento o aprendizaje.

En la tabla 4.1 se muestran los porcentajes de **error** obtenidos para cada uno de los métodos en los diferentes pasos (PASOS 1, 2 y 3) y en los diferentes conjuntos (S0, S1, S2 y S3). En la tabla aparecen varios grupos de clasificadores, a saber:

- **Métodos híbridos con lógica fuzzy**, constituye el grupo de los seis métodos (H1 a H6) de combinación basados en la lógica fuzzy propuestos en este trabajo según se describe en la sección 3.4
- **Métodos simples sin hibridación**, que hacen referencia a los métodos individuales utilizados como elementos de la combinación, se trata en concreto del clasificador de Bayes (BY), el clasificador Fuzzy (FU) y el clasificador de Parzen (PA).

- **Métodos híbridos clásicos**, se trata de los métodos más comúnmente utilizados en la literatura y descritos en la sección 2.3, a saber: VM, MA, MI y ME.
- **Clasificador híbrido** utilizando exclusivamente los clasificadores Fuzzy y Bayesiano (FB), que constituye el origen de este trabajo de investigación tal y como se describe en la sección 3.2

	% Error	PASO 1		PASO 2		PASO 3	
		S0	S1	S0	S2	S0	S3
<b>Métodos Híbridos con lógica Fuzzy</b>  (propuestos en este trabajo)	<b>H1</b>	36.2	37.3	36.1	34.8	32.3	31.2
	<b>H2</b>	29.6	30.4	29.2	28.8	26.8	25.2
	<b>H3</b>	27.0	27.5	24.1	20.9	22.1	19.8
	<b>H4</b>	24.1	27.1	22.6	21.1	18.9	19.9
	<b>H5</b>	37.3	38.9	36.4	34.6	32.8	31.9
	<b>H6</b>	25.2	29.3	23.2	22.6	20.1	21.2
<b>Métodos simples sin hibridación</b>	<b>BY</b>	30.2	29.1	26.1	26.2	21.4	20.4
	<b>FU</b>	32.1	30.2	24.5	27.4	22.2	21.9
	<b>PA</b>	33.5	33.6	32.5	31.4	27.3	26.4
<b>Métodos Híbridos clásicos</b>	<b>VM</b>	29.3	30.1	26.3	25.8	23.9	21.6
	<b>MA</b>	30.4	29.6	25.1	26.9	22.0	21.1
	<b>MI</b>	36.4	36.1	31.4	34.3	29.9	28.3
	<b>ME</b>	28.6	28.3	24.8	25.8	22.6	22.3
<b>Híbrido Fuzzy + Bayes (Figura 3.2)</b>	<b>FB</b>	26.1	29.4	23.9	22.9	20.5	21.3

Tabla 4.1 Porcentajes de error obtenidos por los diferentes métodos en los tres pasos y para los tres tipos de conjuntos S1, S2 y S3

Con el fin de visualizar mejor los resultados de la tabla 4.1, a continuación se muestran a través de las figuras 4.8 a 4.10 la evolución del error a lo largo de los tres pasos especificados para el conjunto S0 y según los grupos de métodos descritos en la tabla 4.1.

En la figura 4.8 se muestra el comportamiento de los métodos híbridos H1, H2, H3, H4, H5 y H6, observándose que H4 muestra el mejor comportamiento frente al resto en

términos de menor error en la clasificación. Por tanto, este es el método seleccionado para llevar a cabo su comparación frente al resto de métodos.

En la figura 4.9 se muestra el comportamiento de los clasificadores simples (Fuzzy, Bayes y Parzen) frente al método híbrido H4.

Mientras que en la figura 4.10 se muestra el comportamiento de H4 frente a los clasificadores híbridos clásicos (Votador mayoritario, Regla del máximo, regla del mínimo, regla de la media).

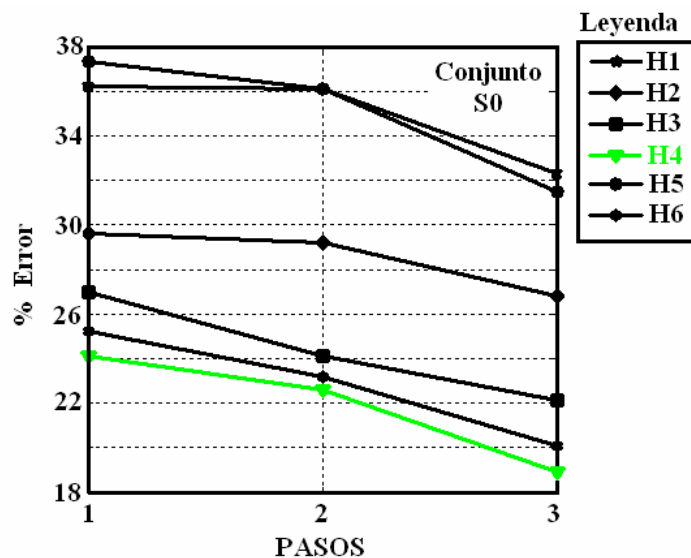


Figura 4.8 Comportamiento de los métodos híbridos propuestos en este trabajo para el conjunto S0 a lo largo de los PASOS 1 a 3

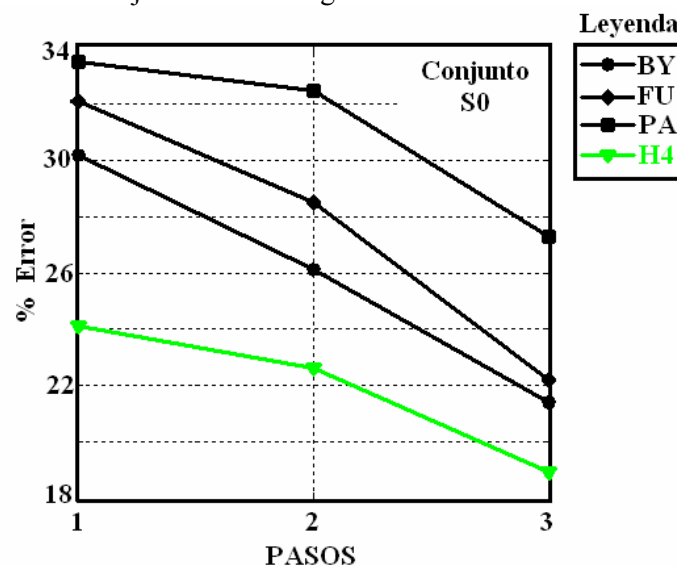


Figura 4.9 Comportamiento de los métodos simples (Bayes, Fuzzy y Parzen) frente al método H4 para el conjunto S0 a lo largo de los PASOS 1 a 3

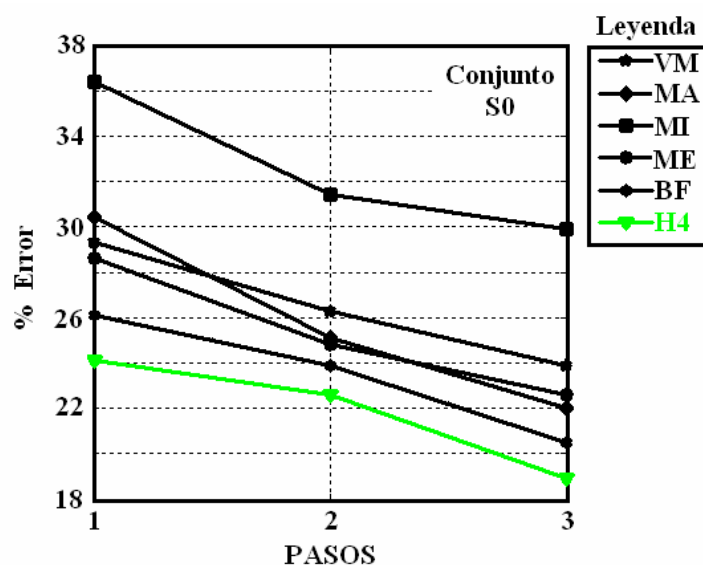


Figura 4.10 Comportamiento de los métodos clásicos de combinación de clasificadores (Votador mayoritario, regla del máximo, mínimo, media) incluyendo la fusión de Fuzzy y Bayes frente al método H4 para el conjunto S0 a lo largo de los PASOS 1 a 3

A la vista de los resultados mostrados en la tabla 4.1, algunos de ellos visualizados gráficamente en las figuras 4.8 a 4.10 se pueden extraer las siguientes conclusiones:

1. Los mejores resultados se obtienen con el método híbrido H4, que muestra el mejor comportamiento, en términos de porcentaje de errores, frente a todos los demás.
2. Se verifica cómo a medida que el número de muestras aumenta, los resultados son mejores (menor número de errores). Eso se visualiza en las tres gráficas al comprobar que los errores para la clasificación del conjunto S0 son mejores en el paso 2 que en el paso 1, y en el 3 que en el 2, es decir cuando se han utilizado más muestras.
3. Los clasificadores híbridos mejoran los resultados de la clasificación, tal y como se ha puesto de manifiesto en numerosas referencias en la literatura.
4. La lógica del producto aparece como una lógica aceptable como lo demuestra el hecho de los mejores resultados obtenidos mediante H4 e incluso mediante BF, que en realidad realiza un producto de los valores de pertenencia proporcionados por el clasificador Fuzzy con las probabilidades obtenidas mediante Bayes.
5. Los porcentajes de error obtenidos no constituyen el óptimo, ya que en el mejor de los casos estamos alrededor del 18%.



# Capítulo 5

## 5. Conclusiones

### 5.1. Sobre resultados

En este trabajo se propone un clasificador no supervisado híbrido. Es bien sabido que las técnicas supervisadas producen mejores resultados que las no supervisadas, lo cual es una consecuencia lógica del proceso de supervisión mismo. El diseño de un clasificador no supervisado a partir de clasificadores supervisados constituye en sí mismo una aportación. La gran ventaja de los métodos no supervisados estriba en su naturaleza automática. Es decir se trata de un mecanismo de clasificación, que puede incorporarse a cualquier proceso de clasificación en el que la intervención de un supervisor no sea factible o no aconsejable.

Por otro lado, la hibridación ha sido una técnica ampliamente utilizada en la literatura. Para ello se ha recurrido a diferentes formulaciones y paradigmas. Los resultados obtenidos con las técnicas híbridas han sido normalmente mejores que con los métodos simples. Esto es así porque gracias a la hibridación es posible quedarse con las ventajas y rechazar los inconvenientes de los clasificadores individuales. Tratando de mejorar siempre los resultados, en este trabajo se ha recurrido a la lógica fuzzy como la metodología adecuada para llevar a cabo una combinación de clasificadores. Esto es así porque la mencionada lógica es capaz de englobar incluso las técnicas más comúnmente utilizadas a la vez que amplía el rango de operabilidad.

Los métodos híbridos propuestos están basados en una sólida teoría de operaciones algebraicas entre conjuntos borrosos. Se han analizado las características de generalizar las operaciones algebraicas clásicas, encontrando inconvenientes en la lógica de Zadeh y Lukasiewicz, por lo que este trabajo se ha centrado de forma innovadora en la lógica del producto. Otros métodos existentes (máximo, mínimo) son un caso particular del enfoque

ofrecido en este trabajo, que además, ha permitido combinar diferentes operaciones entre los tres métodos híbridos.

Las dos principales aportaciones mencionadas anteriormente junto con los resultados obtenidos hacen del presente trabajo una buena herramienta para la clasificación de texturas naturales.

Uno de los problemas encontrados en la clasificación de texturas naturales como las procesadas en el presente trabajo proviene del hecho de no disponer de una base de datos de imágenes de estas características. Esto implica que los resultados de error obtenidos en la clasificación no sean todo lo buenos que sería de esperar. No obstante, esto no anula la validez del método que se ha diseñado, ya que se ha verificado un comportamiento muy apropiado, que supera al de aquellos clasificadores suficientemente contrastados y verificados en la literatura, tanto a nivel individual como a nivel de combinación de los mismos.

## **5.2. Cumplimiento de los objetivos propuestos**

En relación a los objetivos generales expresados en la sección 1.2, cabe añadir que se han cumplido las expectativas previstas.

Respecto de los específicos, su grado de cumplimiento se concreta como sigue:

1. Se han estudiado exactamente los siguientes métodos de clasificación: Fuzzy, Bayes, Parzen, Mapas auto-organizativos, cuantización vectorial y redes neuronales
2. Se han utilizado imágenes aéreas de texturas
3. Se ha propuesto e implementado un método híbrido con una aportación novedosa, hasta donde hemos podido investigar
4. Se proponen líneas de investigación futuras
5. Se han materializado los resultados en forma de ponencias enviadas a varios congresos nacionales e internacionales.

### 5.3. Trabajos futuros

- Análisis de otras características, tales como HSI u otras combinaciones
- Los errores de clasificación podrían mejorarse
- Elaboración de una base de datos o ground-truth más elaborada partiendo de la generación de datos sintéticos y búsquedas en internet
- Estudio de los problemas relacionados con la iluminación, ya que las imágenes se capturan en diferentes días y por tanto bajo diferentes condiciones de iluminación.
- Estudio de resultados con otros métodos híbridos obtenidos a partir de otras operaciones algebraicas y otros métodos de agregación, o utilizando otros operadores de distintas lógicas borrosas.
- Estudio de un método propio borroso de clasificación supervisada utilizando indistinguibilidades que asignen grados de similaridad de píxeles a clústeres usando negaciones de distancias generalizadas.

### 5.4. Divulgación de resultados

Como consecuencia de la investigación realizada en el ámbito de la clasificación de texturas son varios los trabajos sometidos a procesos de revisión como se indica a continuación

1. María Guijarro, Raquel Abreu, Gonzalo Pajares y Luis Garmendia. Design of a Hybrid Classifier for Natural Textures in Images from the Bayesian and Fuzzy Paradigms. IEEE International Symposium on Intelligent Signal Processing (WISP07).
2. María Guijarro, Raquel Abreu, Gonzalo Pajares, A new unsupervised hybrid classifier for Natural Texturas in images, In Second Internacional Workshop on Hybrid Artificial Intelligence Systems (HAIS07).
3. María Guijarro, Raquel Abreu, Gonzalo Pajares. On Combining Learning Vector Quantization and the Bayesian classifiers for natural textured images, En Segundo Congreso Español de Informática (Minería de Datos y Aprendizaje), CEDI07.

Para estos trabajos se ha establecido un proceso de colaboración con el Departamento de Informática y Automática de la UNED, donde se están estudiando otros métodos de transformación del color

## 6. Bibliografía

B. Balasko, J. Abonyi and B. Feil, Fuzzy Clustering and Data Analysis Toolbox for Use with Matlab, Veszprem University, Hungary

URL:<http://www.fmt.vein.hu/softcomp/fclusttoolbox/FuzzyClusteringToolbox.pdf>

BME Texture Database (disponible on-line [http://bme.t.u-tokyo.ac.jp/tech\\_ref/texture\\_DB/search.asp](http://bme.t.u-tokyo.ac.jp/tech_ref/texture_DB/search.asp))

J.C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Kluwer, Plenum Press, New York, 1981.

Brodatz Texture Database (disponible on-line <http://www.ux.uis.no/~tranden/brodatz.html>)

J.C.W. Chan, N. Laporte and R.S. Defries, "Texture Classification of logged forest in tropical Africa using machine-learning algorithms," *Int. J. Remote Sensing*, vol. 24, n° 6, pp. 1401-1407, March 2003.

D. Deng and J. Zhang, "Combining Multiple Precision-Boosted Classifiers for Indoor-Outdoor Scene Classification," Internal Report N° 2006/09, Dpt. Information Science, University of Otago, Dunedin, New Zeland, 2006.

A. Drimbarean and P.F. Whelan, "Experiments in colour texture analysis," *Pattern Recognition Letters*, vol. 22, n° 4, pp. 1161-1167, April 2003.

R.O. Duda, P.E. Hart and D.S. Stork, *Pattern Classification*, Wiley, 2000.

R.P.W. Duin, "On the choice of smoothing parameters for Parzen estimators of probability density functions," *IEEE Trans. Comput.*, C-25, pp. 1175-1179.

A. de la Escalera, "Visión por Computador: fundamentos y métodos", Prentice-Hall, 2001

F. del Frate, F. Pacifici, G. Schiavon, C. Solimini, "Use of Neural Networks for Automatic Classification from High-Resolution Images," *IEEE Trans. Geoscience and Remote Sensing*, vol. 45, n° 4, pp. 800- 809, April 2007.

K. Fukunaga and M.D. Hummels, "Bayes error estimation using Parzen and K-NN procedures," *IEEE Trans Pattern Anal. Machine Intelligence*, vol. 9, n° 5, pp. 634-643, May 1987.

G. Giacinto, F. Roli and L. Bruzzone, "Combination of neural and statistical algorithms for supervised classification of remote-sensing image," *Pattern Recognition Letters*, vol. 21, no. 5, pp. 385-397, May 2000.

R.C. Gonzalez and R.E. Woods, "Digital Image Processing", Prentice-Hall, 2002

J. Grim, J. Kittler, P. Pudil and P. Somol "Multiple Classifier Fusion in Probabilistic Neural Networks," *Pattern Analysis and Applications*, 5, 221-233, 2002.

M. Hanmandlu, V.K. Madasu and S. Vasikarla, "A Fuzzy Approach to Texture Segmentation," in *Proc. of the IEEE International Conference on Information Technology: Coding and Computing (ITCC'04)*, The Orleans, Las Vegas, Nevada, USA, April. 2004, pp. 636-642.

T.K. Ho, "A Data Complexity Analysis of Comparative Advantages of Decision Forest Constructors," *Pattern Analysis and Applications*, 5, 102-112, 2002.

A.K. Jain and M.D. Ramaswami, "Classifier design with Parzen windows," in *Pattern Recognition and Artificial Intelligence toward an integration*, E.S. Geselma and L.N. Kanal (Eds.) Amsterdam, The Netherlands, 1988, pp. 211-218.

D.W. Kim, K.H. Lee and D. Lee, "Fuzzy Clúster validation index based on inter-clúster proximity," *Pattern Recognition Letters*, vol. 24, pp. 2561-2574, November 2003.

J. Kittler, M. Hatef, R.P.W. Duin and J. Matas, "On Combining Classifiers" *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226-239, March 1998.

S. Kumar, J. Ghosh and M.M. Crawford, "Hierarchical Fusion of Multiple Classifiers for Hyperspectral Data Analysis," *Pattern Analysis and Applications*, 5, 210-220, 2002.

S. Kumar, J. Ghosh and M.M. Crawford, "Best-bases feature extraction for pairwise classification of hyperspectral data," *IEEE Trans. Geoscience and Remote Sensing*, 39(7), 1368-1379, 2001.

L.I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*, Wiley, 2004.

P. Maillard, Comparing Texture Analysis Methods through Classification, *Photogrammetric Engineering and Remote Sensing*, vol. 69, nº 4, pp. 357-367, April, 2003.

J.M. Martínez-Otzeta, Combinación de Clasificadores. En Aprendizaje Automático: conceptos básicos y avanzados (Serra, Ed.), Prentice-Hall, 2006.

G. Pajares y J.M. de la Cruz, "Visión por Computador: Imágenes Digitales y Aplicaciones", RA-MA, 2001.

G. Pajares, V. Moreno y J. M. de la Cruz, Clasificación de texturas mediante redes neuronales, *Ingeniería Civil. Centro de Estudios y Experimentación de Obras Públicas (CEDEX).- Ministerio de Fomento*, nº 123, pp. 61-69, (2001).

G. Pajares and J. M. Cruz, Clasificación de Texturas Naturales mediante K-Means, *Revista Electrónica de Visión por Computador* <http://revc.uab.es/revista/06/>, no. 6, pp. 1-18, Ed. Centre de Visió per Computador, Universitat Autònoma Barcelona, 2002.

G. Pajares, J. M. de la Cruz y V. Moreno, Clasificación de texturas naturales mediante agrupamiento borroso, *Ingeniería Civil. Centro de Estudios y Experimentación de Obras Públicas (CEDEX).- Ministerio de Fomento*, nº 127, pp. 83-89, 2002.

D. Partridge and N. Griffith, "Multiple Classifier Systems: Software Engineered, Automatically Modular Leading to a Taxonomic Overview," *Pattern Analysis and Applications*, 5, 180-188, 2002.

E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Statist.*, vol. 33, pp. 1065-1076, 1962.

D. Puig and M.A. García, "Automatic texture feature selection for image pixel classification," *Pattern Recognition*, vol. 39, nº 11, pp. 1996-2009, November 2006.

T. Randen and J.H. Husøy, "Filtering for Texture Classification: A Comparative Study," *IEEE Trans. Pattern Analysis Machine Intell.*, vol. 21, n° 4, pp. 291-310, April, 1999.

R. Rud, M. Shoshany, V. Alchanatis and Y. Cohen, "Application of spectral features' ratios for improving classification in partially calibrated hyperspectral imagery: a case study of separating Mediterranean vegetation species", *Journal Real-Time Image Processing*, vol. 1, pp. 143-152, 2006.

B. Schweizer, A. Sklar. Probabilistic Metric Space. 1983.

G. Smith, I. Burns, "Measuring texture classification algorithms", *Pattern Recognition Letters*, vol. 18, pp. 1495-1501, 1997.

R.M. Valdovinos, J. S. Sánchez and R. Barandela. "Dynamic and Static weighing in classifier fusion," in *Pattern Recognition and Image Analysis*, Lecture Notes in Computer Science (J.S. Marques, N. Pérez de la Blanca and P. Pina, Eds.), Springer-Verlag, Berlin. 2005, pp. 59-66.

VisTex (disponible on-line

<http://vismod.media.mit.edu/vismod/imagery/VisionTexture/information.html>)

T. Wagner , "Texture Analysis". In *Handbook of Computer Vision and Applications* (B. Jähne, H. Haußecker and P. Geißler), Academic Press, vol. 2 (Signal Processing and Pattern Recognition), 1999.

H. Yu, M. Li, H.J. Zhang and J. Feng, "Color texture moments for content-based image retrieval". In *Proc. International Conference on Image Processing*, vol. 3, pp. 24-28, 2002.

H.J. Zimmermann, *Fuzzy Set Theory and its Applications*, Kluwer Academic Publishers, Norwell, 1991.



## 7. Anexo I

# Ventanas de diálogo: aprendizaje y clasificación

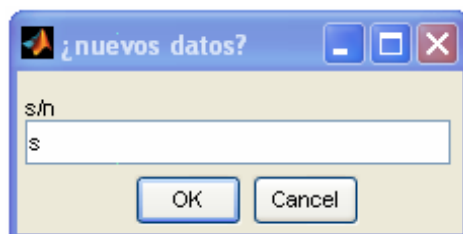
A continuación se presenta la aplicación desarrollada para llevar a cabo los experimentos presentados en el presente trabajo. En dicha aplicación aparecen diálogos de ventanas correspondientes a otros procesos de clasificación que se encuentran actualmente bajo estudio como ampliación del trabajo que se presenta. Hemos preferido mantenerlos para dar una idea más clara de las líneas de investigación.

Se distinguen dos procesos, según el esquema de la figura 3.1, a saber: **off-line o aprendizaje** y **on-line o clasificación**.

### APRENDIZAJE

El proceso de aprendizaje se lleva a cabo mediante el script **Aprendizaje.m** cuya ejecución se realiza desde la línea de comandos. Los pasos que se describen a continuación se identifican en el código, exactamente con esta referencia.

**Paso 1:** cargar datos (si existen de la base de Conocimiento), estos datos se cargan sobre la variable  $X$  cuya dimensión es  $N \times 3$ , conteniendo valores de color de los píxeles de las muestras. Si se contesta que **n** se cargarán sólo los datos existentes en la BC



**Paso 2:** si en el caso anterior se selecciona **si** (s, opción por defecto), se comienza a ejecutar un bucle de forma que en cada ejecución del bucle abre una nueva imagen que ha sido previamente seleccionada y numerada como imagen de entrenamiento con la

nomenclatura **IEntrenamiento\_xx.bmp**, donde xx es un valor numérico que se corresponde con el número de iteración. La imagen cargada se muestra en pantalla.

Tras la apertura de una imagen todavía se pregunta si se quiere seleccionar las muestras con el ratón de forma que se seleccionen píxeles de forma controlada o por el contrario que todos los valores de la imagen sean cargados como muestras. El diálogo es el siguiente:



En caso de contestar **si** (s, opción por defecto), se van seleccionando las muestras con el botón izquierdo, pulsando una vez, cuando se quiera dejar de capturar más muestras se pulsa el botón derecho. Sobre la imagen aparece una marca indicando las muestras que se han seleccionado.

Ante cualquiera de las dos opciones se vuelve a preguntar si se desean capturar más muestras, siendo posible de nuevo volver a seleccionar o bien con el ratón o bien toda la imagen. En una iteración se puede seleccionar con el ratón y en la siguiente toda la imagen o viceversa sin que haya ninguna restricción al respecto.

La ventana de diálogo que aparece es la siguiente con la opción **si** (s, por defecto). Cuando se opta por **no** (n), se termina el bucle de captura y por tanto la toma de muestras.

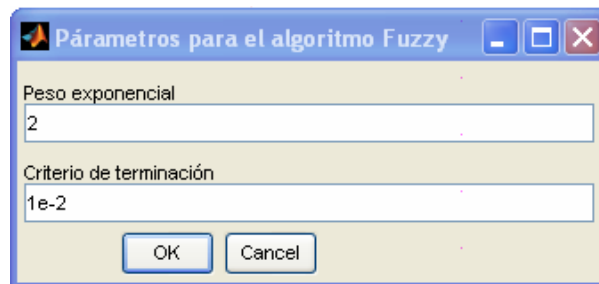


Los nuevos datos se añaden a la variable  $X$  de muestras previamente cargada procedente de la BC, si no hubiera muestras previas,  $X$  contendría las muestras seleccionadas.

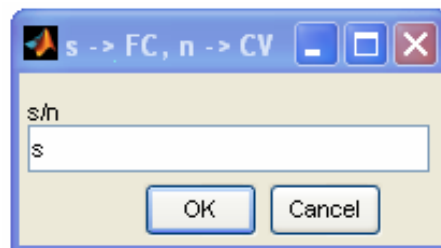
Una vez capturadas todas las muestras, éstas se almacenan en la BC, con el fin de tenerlas disponibles para futuros procesos de aprendizaje.

Las muestras obtenidas son valores de los píxeles en el rango  $[0,255]$ , tras su captura se procede a su normalización al rango  $[0,1]$ .

**Paso 3:** se inicia el proceso de partición de los datos, que como se ha mencionado previamente se basa en uno de los dos algoritmos **FC** o **CV**. El primer diálogo que se muestra a continuación relativo a dos parámetros requerido por el algoritmo **FC**, ya que son necesarios para el proceso de validación posterior independientemente del método que se elija (**FC** o **CV**), aunque el **CV** no los requiera. Estos parámetros son el peso exponencial ( $m$ ) y el criterio de terminación (o error de tolerancia). Los valores por defecto son los que aparecen en la ventana.



Seguidamente se muestra la siguiente ventana de forma que si se elige **si** (s, por defecto) se realizará la partición con el método **FC** y si es **no** (n) será con el **CV**.



La respuesta **si** implica que es necesario proporcionar el número de clústeres para realizar la clasificación mediante **FC** apareciendo en consecuencia la siguiente ventana,

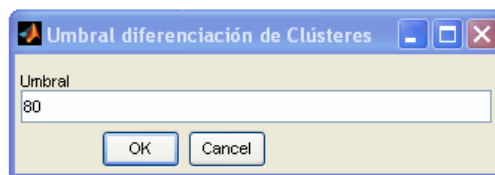


En este caso se llama a la función **fuzzy\_clustering** con las muestras  $X$  y los parámetros  $c$ ,  $m$ , error, VEstimados (que en este caso no tiene efecto ninguno) y DatosClasificados = false, porque en esta fase los datos no han sido todavía clasificados. Este último parámetro se utiliza para identificar si los centros deben inicializarse aleatoriamente o no al tener una clasificación previa (como se verá posteriormente). La función devuelve una matriz  $v$  conteniendo los centros de los clústeres, esta matriz es de dimensión  $c \times 3$  siendo  $c$  el

número de clústeres y 3 el número de componentes de cada centro, como estamos tratando con datos de píxeles R,G, B de ahí el valor 3. Por tanto, cada fila se corresponde con el centro de cada clúster. También devuelve una matriz  $U$  que indica el grado de pertenencia de cada una de las muestras a cada uno de los clústeres. Esta matriz será por tanto es de dimensión  $N \times c$ , ya que hay  $N$  muestras y  $c$  clústeres. La matriz  $v$  está normalizada al rango  $[0,1]$ , tras este paso se procede a la desnormalización ya que los datos originales que van a agruparse están sin normalizar. El resto de parámetros devueltos son informativos y no se requieren de momento.

Seguidamente, se procesan todas las muestras identificando a qué clase corresponde el mayor grado de pertenencia y asignado la muestra a la clase con ese mayor grado. De esta forma se obtienen las muestras y el número de las mismas asignadas a cada clase. Esto se almacena en la variable DATOS\_CLASES.

Si hubiésemos elegido la opción **no**, se ejecutaría el algoritmo **CV** para lo cual se requiere el umbral  $T$  de separación de los clústeres, que se solicita a través de la ventana de diálogo siguiente,



Seguidamente, se procesan todas las muestras identificando a qué clase corresponde el mayor grado de pertenencia y asignado la muestra a la clase con ese mayor grado. De esta forma se obtienen las muestras y el número de las mismas asignadas a cada clase. Esto se almacena en la variable DATOS\_CLASES.

**Paso 4:** validación de resultados. A partir de los datos clasificados contenidos en DATOS\_CLASES se procede a validar la partición obtenida previamente. Se inicia el proceso reconstruyendo el vector de datos  $X$ , para ser pasados a la función **fuzzy\_clustering** la diferencia con la anterior ejecución estriba en que ahora se inicializan los vectores representativos de cada clase tendiendo en cuenta que ya se dispone de una clasificación previa. Esto se lleva a cabo indicando que DatosClasificados = true.

Tras la normalización al rango  $[0,1]$  tanto de las muestras  $X$  como de los vectores representativos de cada clase se procede a llamar a la función **fuzzy\_clustering**, que devuelve los nuevos vectores representativos de cada clase a través de la matriz  $(v)$ , la

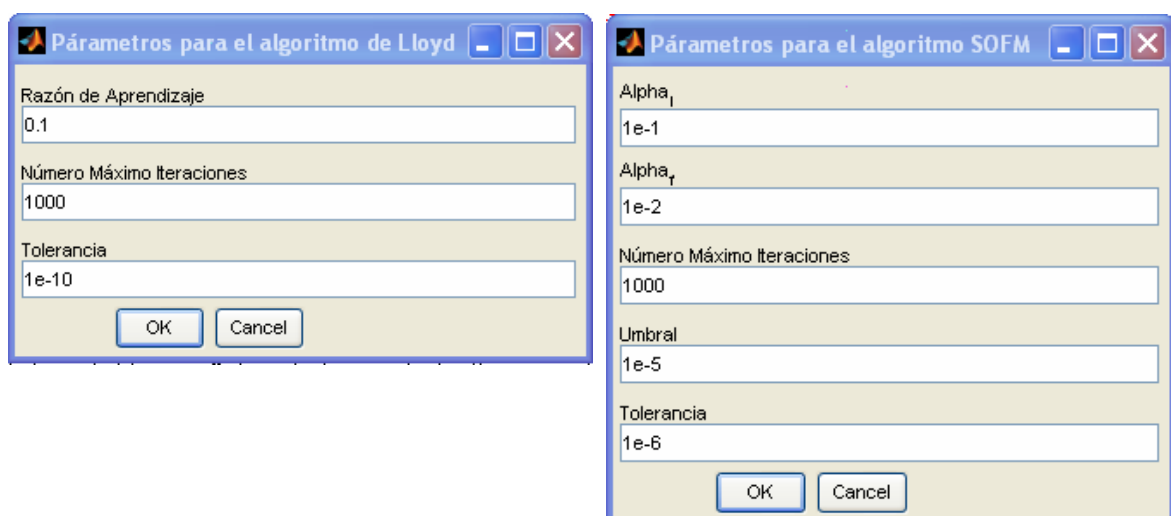
matriz de grados de pertenencia ( $U$ ) y una distancia, que se utiliza en el método de validación fuzzy. Dependiendo de cual sea el valor del parámetro coeficiente, así se obtienen unos u otros coeficientes de validación fuzzy. Estos coeficientes son los proporcionados en las referencias de métodos y se obtienen en `CvalidezFuzzy` (en la implementación actual sólo se utiliza el coeficiente PC, en implementaciones futuras hay que combinar los restantes con los derivados del clasificador de Bayes).

Se procede a la desnormalización de  $v$  para ser almacenada posteriormente en la BC.

Seguidamente se procede a realizar la clasificación mediante el clasificador **Bayesian** al que se le pasan los datos previamente clasificados como pertenecientes a las diversas clases y devuelve la media y la matriz de covarianza asociadas con cada una de las clases en la variable `DATOS_CLASES`.

El paso final del proceso de validación consiste en el cómputo de los valores de validación correspondientes a la agrupación Bayesiana a través de la función **validezBayesian** que devuelve los coeficientes de Divergencia, coseno y Jeffries-Matusita entre las clases (de momento estos valores no se evalúan).

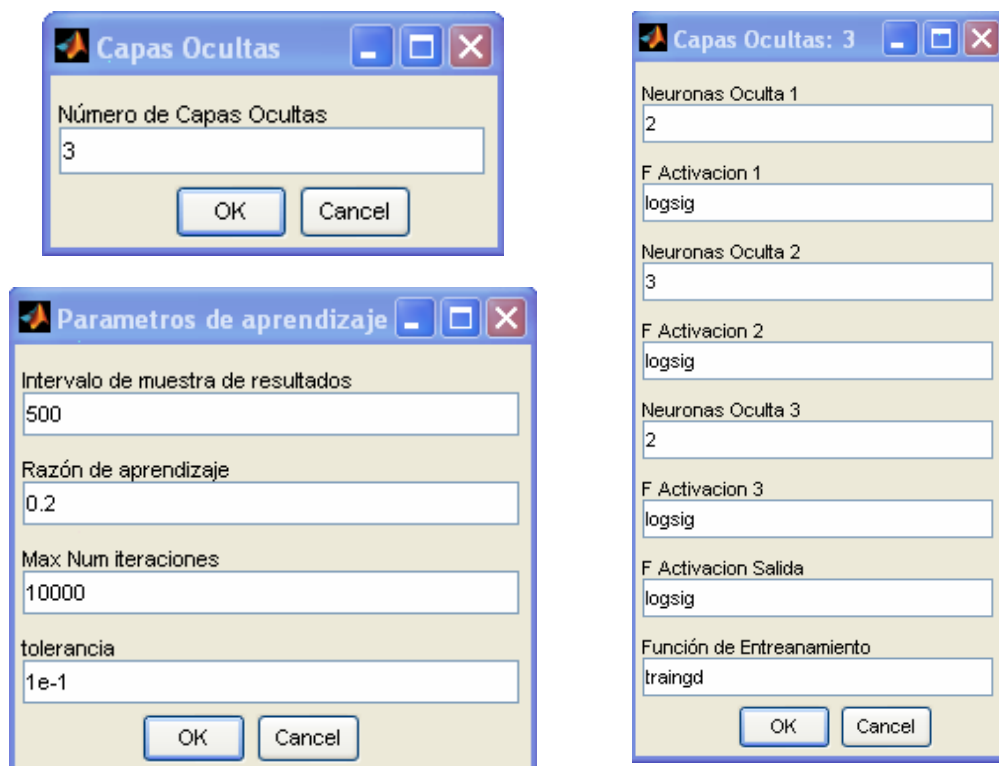
**Paso 5:** si la validación no ha sido satisfactoria, se procede a realizar una nueva partición, no obstante si ésta ha resultado ser satisfactoria, se continúa al proceso de aprendizaje mediante los métodos **MA,CL** preguntándose sobre los parámetros requeridos a través de las siguientes ventanas de diálogo respectivamente. En ambos procedimientos los parámetros aprendidos son los centros de las clases, que serán almacenados en la BC



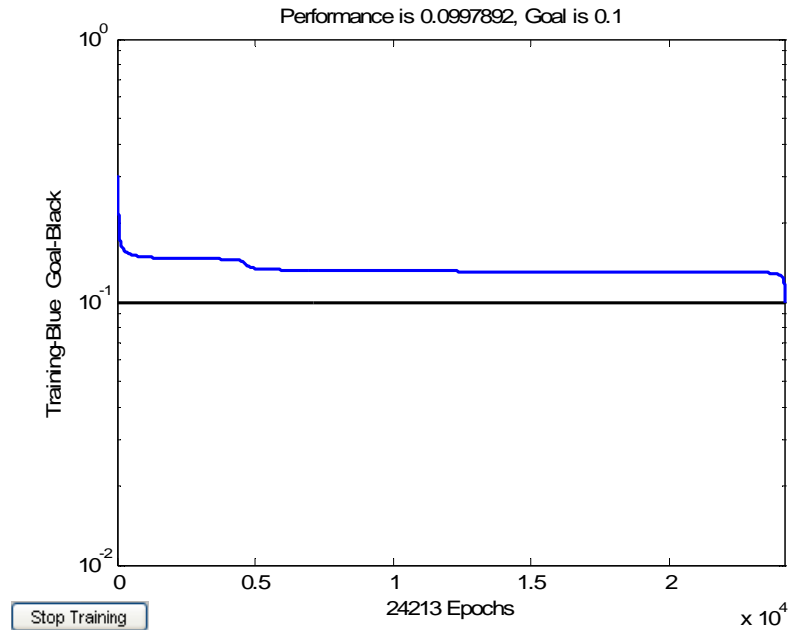
Dentro de la fase de aprendizaje se incluye un último proceso, que es el clasificador basado en la red neuronal **CR**. La primera ventana de diálogo que aparece pregunta sobre el número de capas ocultas que va a tener la red mediante el procedimiento **TopologiaRedCB** y mediante las ventanas de diálogo 1, 2 y 3.

Mediante la función **RetroPropagacion** se crea la red con la función **newff** y finalmente se procede al entrenamiento con la función **train**. Tras este proceso la red creada modifica (aprende) los pesos y se guarda en la BC exactamente la red entrenada como tal.

Existe un proceso final para verificar que la red ha aprendido correctamente, que consiste en que suministrando las muestras con las que ha aprendido debe generar exactamente los patrones de entrenamiento con los que se ha entrenado. Ese último proceso se lleva a cabo mediante las funciones **GenerarPatronObjetivo** y **sim**.



La red inicia su proceso de entrenamiento mostrando la siguiente información,

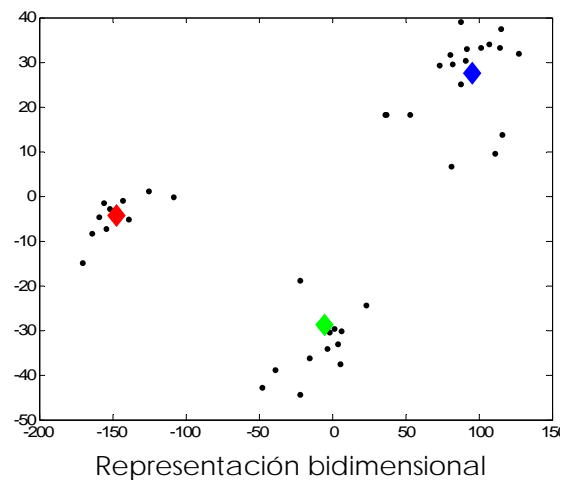
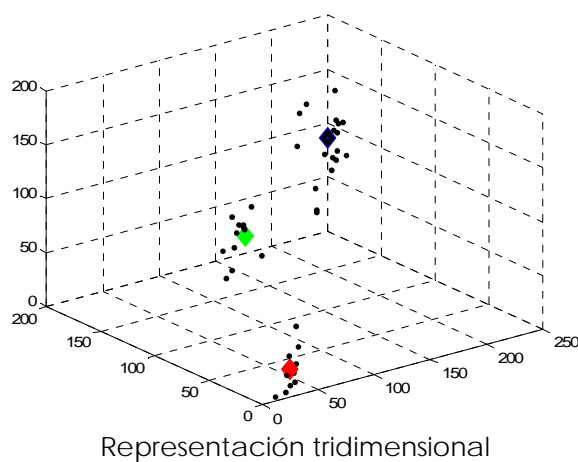


**Paso 6:** finalmente, se almacenan en la BC los datos aprendidos en el fichero **ParamAprendizaje**

**Paso 7:** verificación del aprendizaje de la red

**Paso 8:** representación tridimensional y bidimensional de los datos originales y los centros Fuzzy. En el caso bidimensional se utiliza la proyección de **Componentes Principales** para reducir en una dimensión (de tres a dos) los datos y centros.

Se muestran los resultados que aparecen en la imagen, donde los datos aparecen representados en los puntos negros y los centros como rombos coloreados.



## CLASIFICACIÓN

El proceso de aprendizaje se lleva a cabo mediante el script **Clasificador.m** cuya ejecución se realiza desde la línea de comandos. Los pasos que se describen a continuación se identifican en el código, exactamente con esta referencia.

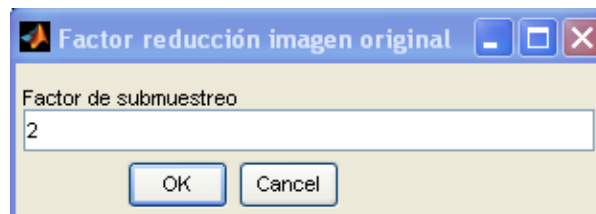
**Paso 1:** carga de parámetros desde la Base de Conocimiento (BC). Estos parámetros son los almacenados durante la fase de Aprendizaje previa. Su carga se realiza mediante el comando

**load ParamAprendizaje**

Se recupera toda la información almacenada en la estructura DATOS\_CLASES

**Paso 2:** generación del mapa de colores para la representación gráfica de las clases identificadas. Se realiza mediante la función **MapaColores** generando una disposición de los colores a representar. Este mapa es el mostrado en la sección Resultados obtenidos

**Paso 3:** se pregunta por si se desea reducir la dimensionalidad de la imagen, ya que el proceso es computacionalmente costoso. El diálogo mostrado es el que aparece a continuación

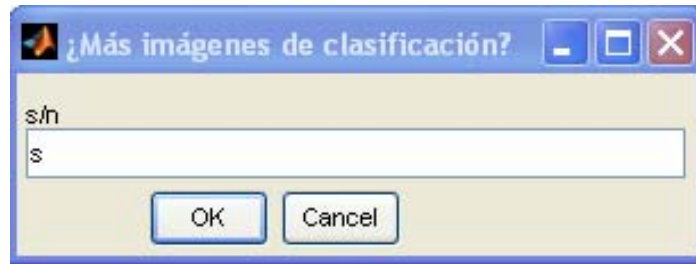


**Paso 4:** se inicia un proceso iterativo para proceder a la carga de nuevas imágenes con la nomenclatura **IClasificacion\_xx.bmp**, donde xx es un valor numérico que se corresponde con el número de iteración. La imagen cargada se muestra en pantalla.

Estas imágenes se procesan píxel a píxel, obteniendo los valores de cada píxel a través de la variable *píxel*.

Tras el procesamiento de cada imagen se pregunta sobre si se desea realizar más clasificaciones o no mediante el siguiente diálogo, de forma que si se contesta **si** (s, por defecto) se clasifica una nueva imagen.





El resultado de este proceso es el mostrar por pantalla las imágenes clasificadas y los resultados de la clasificación



## 8. Anexo II

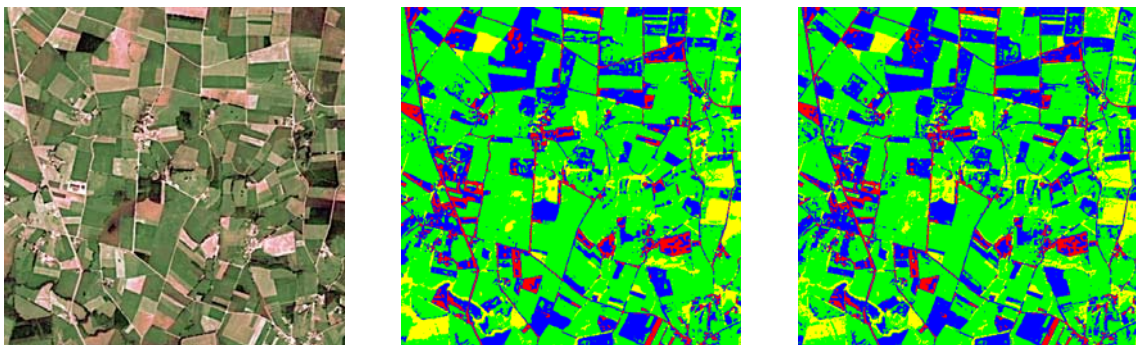
A continuación se muestran cuatro ejemplos de imágenes clasificadas por los métodos que se indican seguidamente.

En la columna de la izquierda aparece la imagen original, en la central la imagen clasificada por el método híbrido H4 descrito en la sección 3.4 a tenor de los resultados derivados en el capítulo cuatro. En la tercera columna los resultados obtenidos con los métodos Fuzzy, Bayes, Parzen e híbrido Fuzzy + Bayes.

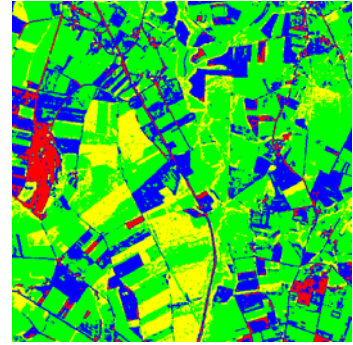
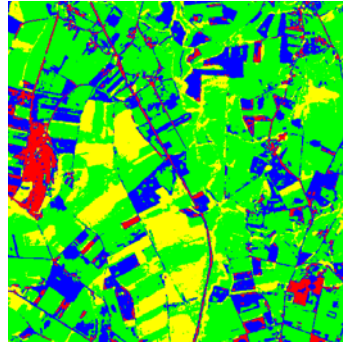
Cualitativamente los resultados entre H4 y el resto de los métodos no parecen diferir considerablemente, como se observa por la mera inspección visual. No obstante, cuando se analizan de forma cuantitativa, mediante el análisis de errores del capítulo cuatro, las diferencias se ponen de manifiesto.

Por otro lado, según ese mismo análisis, los errores de clasificación se sitúan alrededor del 20% en el mejor de los casos, siendo el método H4 el que mejores resultados obtiene.

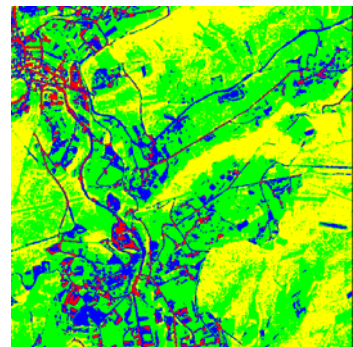
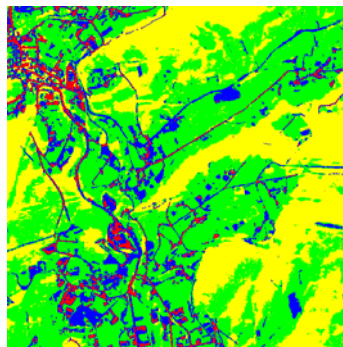
Ese porcentaje dista todavía bastante de ser el óptimo, esto justifica el estudio de futuras investigaciones en el intento de mejora de estos resultados de clasificación. La propia mejora obtenida por el método híbrido H4 justifica igualmente la investigación presente.



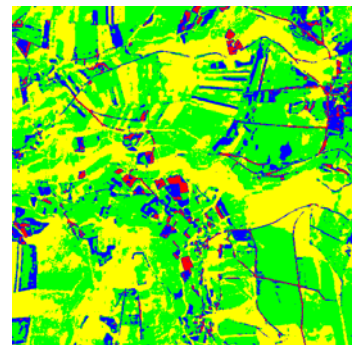
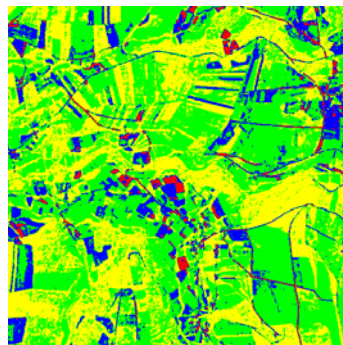
Ejemplo 1



Ejemplo 2



Ejemplo 3



Ejemplo 4